Bacterial Foraging Based Channel Equalizers

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Technology in 'Telematics and Signal Processing'

By

Raghuveer Allamneni

Department of Electronics and Communication Engineering
National Institute of Technology
Rourkela
Orissa

May-2006

CERTIFICATE

This is to certify that the work in this thesis entitled ***"Bacterial Foraging Based Channel Equalizers"*** by ***Raghuveer Allamneni***, has been carried out under my supervision in partial fulfillment of the requirements for the degree of Master of Technology in **'Telematics and Signal Processing'** during session 2005-2006 in the Department of Electronics and Communication Engineering, National Institute of Technology, Rourkela and this work has not been submitted elsewhere for a degree.

Place:
Dated:

Dr. S.K.Patra
Asst.Professor, Dept. of ECE
National Institute of Technology, Rourkela

ACKNOWLEDGEMENTS

On the submission of my Thesis report of **"Bacterial Foraging Based Channel Equalizers**", I would like to extend my gratitude & my sincere thanks to my supervisor Dr.S.K.Patra, Asst. Professor, Department of Electronics and communication Engineering for his constant motivation and support during the course of my work in the last one year. I truly appreciate and value his esteemed guidance and encouragement from the beginning to the end of this thesis. His knowledge and company at the time of crisis would be remembered lifelong.

I want to thank all my teachers **Dr G.S Rath, Dr G. Panda and Dr K. K. Mahapatra** for providing a solid background for my studies and research thereafter. They have been great sources of inspiration to me and I thank them from the bottom of my heart.

I will be failing in my duty if I do not mention the laboratory staff and administrative staff of this department for their timely help.

I would like to thank all whose direct and indirect support helped me completing my thesis in time.

I would like to thank all those who made my stay in Rourkela an unforgettable and rewarding experience.

Last but not least I would like to thank my parents, who taught me the value of hard work by their own example. I would like to share this moment of happiness with my father and mother. They rendered me enormous support during the whole tenure of my stay in NIT Rourkela.

Raghuveer Allamneni
M.Tech (Telematics & Signal processing)

Abstract

A channel equalizer is one of the most important subsystems in any digital communication receiver. It is also the subsystem that consumes maximum computation time in the receiver. Traditionally maximum-likelihood sequence estimation (MLSE) was the most popular form of equalizer. Owing to non-stationary characteristics of the communication channel MLSE receivers perform poorly. Under these circumstances 'Maximum A-posteriori Probability (MAP)' receivers also called Bayesian receivers perform better.

Natural selection tends to eliminate animals with poor "foraging strategies" and favor the propagation of genes of those animals that have successful foraging strategies since they are more likely to enjoy reproductive success. After many generations, poor foraging strategies are either eliminated or shaped into good ones (redesigned). Logically, such evolutionary principles have led scientists in the field of "foraging theory" to hypothesize that it is appropriate to model the activity of foraging as an optimization process.

This thesis presents an investigation on design of bacterial foraging based channel equalizer for digital communication. Extensive simulation studies shows that the performance of the proposed receiver is close to optimal receiver for variety of channel conditions. The proposed receiver also provides near optimal performance when channel suffers from nonlinearities.

Contents

List of Figures

List of Tables

Acronyms & Abbreviations

ACI	adjacent channel interference
ANN	artificial neural network
ANFF	adaptive neuro-fuzzy filter
ASK	amplitude shift keying
AWGN	additive white Gaussian noise
BER	bit error ratio
BF	bacterial foraging
BPSK	binary phase shift keying
CCI	co-channel interference
DCR	digital cellular radio
DCS	digital communication system
DFE	decision feedback equalizer
DSP	digital signal processing
EA	evolutionary algorithms
FEXT	far end cross talk
FIR	finite impulse response
HDTV	high definition television
i.i.d	independent identically distributed
IIR	infinite impulse response
ISI	inter symbol interference
LAN	local area network
LMS	least mean square
LS	least square
MAP	maximum *a-posteriori* probability
MLP	multi layer perceptron
MLSE	maximum likelihood sequence estimator
MMSE	minimum mean square error
MSE	mean square error

NEXT	near end cross talk
PAM	pulse amplitude modulation
QPSK	quadrature phase shift keying
RBF	radial basis function
RLS	recursive least squares
SNR	signal to noise ratio
TDL	tapped delay line
TDMA	time division multiple access
VLSI	very large scale integration

Legends

a_i	channel impulse response tap weight i.
B	no: of interfering adjacent channels
C_i	scalar channel state i
c_i	channel state i
C_d	channel state matrix
C_d^+	positive channel state matrix
C_d^-	negative channel state matrix
d	equalizer decision delay
$e(k)$	error signal at time index
ε	expected value
$f(x)$	Arbitrary function 5 with variable
$\Im\{r(k)\}$	equalizer decision function with input vector $r(k)$
$h(t)$	impulse response of a channel
$h_{aci-i}(t)$	impulse response of an adjacent channel i
$h_{co-i}(t)$	impulse response of a co-channel i
$h_C(t)$	inpulse response of a physical channel
$h_R(t)$	impulse response of a receiver matched filter
$h_T(t)$	impulse response of a transmitter modulation filter
$h_{TR}(t)$	combined impulse response of transmitter and receiver filters
H	channel matrix
i	arbitrary variable
$i_1, i_2, ..., i_m$	arbitrary variables
j	arbitrary variable

$j_1, j_2, ..., j_m$	arbitrary variables
k	time index
l	arbitrary variable
L	no: of interfering co-channels
m	equalizer order
M	no: of scalar channel states
n_c	no: of channel taps
q	equalizer feedback order
$r(k)$	received scalar sample at time index k
$\mathbf{r}(\mathrm{k})$	received vector sample at time index k
$\hat{r}(k)$	noise free received scalar at time index k
$\hat{r}_{co}(k)$	co-channel component of received signal $r(k)$
$\Re$	one-dimensional space in $\Re$
$\Re^i$	i-dimensional space in $\Re$
$s(k)$	transmitted sample at time index k
$s_0(k)$	transmitted sample at time index k of the desired channel
$s_{aci-i}(k)$	transmitted sample at time index 2 of adjacent channel i, $1 \le i \le B$
$s_i(k)$	transmitted sample at time index 2 of interfering channel i, $1 \le i \le L$
$S(k)$	transmitted signal vector at time index k
$\hat{s}(k-d)$	estimated sample at time index k with decision delay d
T	time period of transmitted signal
w_i	weight i of a filter/ RBF centre/ neuron / equaliser
$x(k)$	scalar input at time index k
$X(k)$	vector input at time index k

$y(k)$	scalar output at time index k
$\hat{y}(k)$	desired scalar output at time index k
$\eta(k)$	AWGN at time index k
ρ	RBF centres
σ	spread parameter
σ_c^2	channel output signal power
σ_{co}^2	co-channel signal power
σ_s^2	signal power
σ_η^2	channel noise varience
σ_r^2	RBF centre spread parameter
ω_c	channel bandwidth
ω_s	signal bandwidth
$\lvert\cdot\rvert$	absolute distance
$\lVert\cdot\rVert$	Euclidean distance

Chapter-1
Introduction

1.1 Introduction

High speed communications channels are often impaired by channel inter symbol interference (ISI) and additive noise. Adaptive equalizers are required in these communication systems to obtain reliable data transmission. In adaptive equalizers the main constraint is training the equalizer. Many algorithms have been applied to train the equalizer, each having their own advantages and disadvantages. More over the importance of the channel equalization always keeps the research going on to introduce new algorithms to train the equalizer.

Evolutionary algorithms are the emerging techniques in engineering and technology applications. Unlike the conventional algorithms evolutionary algorithms trains the system like a biological system, which means that there is an output (compromised a bit) even if a part(s) of the input is missing. As the evolutionary algorithms train the system like a biological system, it is possible to implement these algorithms using biological computers.

In this thesis work 'Bacterial foraging algorithm', which is a type of evolutionary algorithms, is applied in designing an efficient channel equalizer for high speed digital communications.

The chapter begins with an exposition of the principal motivation behind the work undertaken in this thesis. Following this, section 1.3 provides a brief literature survey on equalization in general and nonlinear equalizers in particular. Section 1.4 outlines the contributions made in this thesis. At the end, section 1.5 presents the thesis layout.

1.2 Motivation of Work

The field of digital data communications has experienced an explosive growth in recent years and its demand reaches at the peak as additional services are being added to existing infrastructure. The telephone networks were originally designed for voice communication but, in recent times, the advances in digital communications using ISDN, data communications with computers, fax, video conferencing etc. have pushed the use of these facilities far beyond the scope of their original intended use. Similarly, introduction of digital cellular radio (DCR) and wireless local area networks (LAN's) have stretched the limited available radio spectrum capacity to the limits it can offer. These advances in digital communications have been made possible by the effective use of the existing communication channels with aid of signal processing techniques. Nevertheless these advances on the existing infrastructure have introduced a host of new unanticipated problems.

The revolution in digital communication techniques can be attributed to the invention of the automatic linear adaptive equalizer in the late 1960's [2]. Adaptive equalizers have gone through many stages of development and refinement in the last 40 years. Early equalizers used linear adaptive filter algorithms with or with out a decision feedback. Alternatively maximum likelihood sequence estimator (MLSE) [3] was implemented using Viterbi algorithm [4, 5].

Both forms of equalizers provided two extremities in-terms of performance achieved and the computational cost involved. The linear adaptive equalizers are simple in structure and easy to train but they suffer from poor performance in severe condition like varying channels as mobile radio channel. On the other hand the infinite memory MLSE provide good performance but at the cost of large computational complexity. Under lower memory constraints MLSE performance also degraded considerably.

As the state of the mobile radio channel always changes and multipath causes time dispersion of the digital information data causing inter-symbol-interference, makes too difficult to detect the actual information at the receiver. It requires adaptive equalizer to adjust its parameters during training to coup of with such fading environment but it needs

large training data or sequences for the linear equalizers and also shows poor performance in case of this mobile radio channel.

The large computational complexity associated with the Viterbi algorithm and poor performance of linear equalizers led to the development of symbol-by-symbol equalizers using the maximum a-posteriori probability (MAP) principle these were also called

Bayesian equalizers[6]. These Bayesian equalizers have been approximated using nonlinear signal processing techniques like Artificial Neural Networks (ANN) [7], radial basis function (RBF) [8,9], recurrent network [9], Kalman filters [10], Fuzzy systems [11] etc. The study of new techniques provides adaptive equalizers which have the advantage of both good performance and comparatively low computational cost. The study of these nonlinear equalizers helps to achieve good performance for mobile channel. Hence different kinds of nonlinear equalizers have been discussed in this thesis.

Evolutionary trains the system like a biological system, so it is possible to implement the systems using biological computers [12].Evolutionary algorithms like genetic algorithms and swarm optimization had already shown encouraging results in solving problems like channel equalization and channel identification in digital communication[13,14,15].The results obtained in harmonic estimation using bacterial foraging [16], have encouraged me to apply the bacterial foraging algorithm to design channel equalizers in digital communication.

1.3 Background Literature Survey

The research in channel equalization started in early 1960's. The earlier equalizers basic theory was of zero forcing equalizers. In 1960 LMS algorithm by Widrow and Hoff [17] shown the way to go for development of adaptive filters used for equalization purposes. In 1965, Lucky [2] used this LMS algorithm to design adaptive channel equalizers. As these equalizers were very simple to design got popularized but very soon their limitations were also revealed in the field of channel equalization. It was seen that these linear equalizers, in spite of best training, could not provide acceptable performance in case of highly dispersive channel and time varying channels. This is due to the fact that linear equalizers treat equalization as an inverse filtering problem whereas equalization can be treated as a pattern classification problem. This led to the investigation of other equalization techniques beginning with MLSE equalizer [3] and its Viterbi implementation [4] in 1970's. Another form of nonlinear equalizer was infinite impulse response (IIR) form of linear adaptive equalizer, where equalizer employs feedback termed as decision feedback equalizer (DFE) [18]. In between 1970's and 1980's , the research works carried out in this field were for the development of faster convergence and computationally efficient algorithms like recursive least square (RLS) algorithm, Kalman filters [10] etc. A review of the development of equalizers till 1985 is available in [19].

In the late 1980's, the beginning of development of field of adaptive neural network (ANN) [7] was seen. The large computational complexity associated with the Viterbi algorithm and poor performance of conventional equalizers with adaptive filters has led to the development of symbol-by-symbol equalizers [6]. These Bayesian equalizers have been approximated using nonlinear signal processing techniques like Artificial Neural Networks (ANN) [7], the multi layer perceptron (MLP) [20] which were computationally more efficient. Another form of nonlinear processors called radial basis function (RBF) [8] were first used for multidimensional functional interpolation. Subsequently these were used for equalization applications [21, 22]. In subsequent years, development of new training algorithms and equalizer structures using ANN [23, 24] and RBF [25] were also developed.

1.4 Thesis Contribution

This section outlines some of major contributions of the study presented in this thesis. In this thesis linear equalizers like LMS as well as nonlinear equalizers like RBF, and Bacterial foraging equalizers have been designed for digital communication applications. The bacterial foraging based equalizers described here are generally classified as nonlinear equalizers. The digital communication problem is discussed first and then need for equalizers is established. Different forms of equalizers are reviewed and with this knowledge of equalizer techniques the bacterial foraging based equalizer and its advantage over the other are described. In the process of evaluation Bit Error rate (BER) has been used as performance measure.

This thesis presents a bacterial foraging implementation of maximum a-posteriori probability (MAP) equalizers based on Bayes's theory. All equalizers developed here are for linear and non-linear channels corrupted with AWGN. It is seen that the advantage provided by the bacterial foraging equalizers in terms of computational complexity and performance gain can provide efficient equalizer design digital communication.

1.5 Thesis outline

Following this introduction the remaining part of the thesis is organized as under: Chapter 2 provides the fundamental concepts of channel equalization and discusses different linear and nonlinear equalization techniques briefly. The channel characteristics that bring out the need for an equalizer in a communication system is also presented. Chapter 3 gives the basic idea of evolutionary algorithms and various types of evolutionary algorithms with emphasis on bacterial foraging. Chapter 4 gives the design of bacterial foraging based equalizers for both linear and non-linear channels. Chapter 5 presents the results obtained for various linear and non-linear channels and the performance is compared with the performance obtained by using LMS and K-means clustering algorithms. Chapter 6 summarizes the work done in this thesis work and points to possible directions for future work.

Chapter – 2
Channel Equalization

2.1 Introduction

This thesis discusses the development of bacterial foraging based channel equalizers for digital communication for a variety of channel impairments. In order to establish the context and need for the work undertaken, it is necessary to discuss the fundamental concepts behind the work. This chapter brings out the need for an adaptive equalizer in a digital communication system (DCS) and describes the classification of adaptive equalizers.

This chapter is organized as follows. Following this introduction, section 2.2 discusses the communication system in general. Section 2.3 discusses the propagation channel model in a DCS, providing the general 'Finite Impulse Response (FIR)' filter model for 'Inter Symbol Interference (ISI)' channels and 'Co Channel Interference (CCI)' channels. Section 2.4 presents the classification of equalizers with emphasis on symbol-by-symbol equalizers.2.5 presents the optimal symbol-by-symbol Bayesian equalizer. Sections 2.6 and 2.7 provide a short overview of developments of linear and nonlinear equalizers respectively, finally section 2.8 presents the concluding remarks.

2.2 Digital Communication System

The Block diagram of a baseband model of a DCS is presented in Fig 2.1. As the analysis of the DCS with all the necessary blocks is very difficult due to complexity associated with all subsystems, communication system are studied in baseband frequency where the encoder, decoder, modulator and the demodulators have been removed. The data source constitutes the signal generation system that generates the information to be transmitted. The work of the encoder in the transmitter is to encode the information bits before transmission so as to provide redundancy in the system. This helps for the correction of errors at the receiver end. Some of typical coding schemes are convolution codes, block codes and grey codes[26]. The digital data transmission requires very large bandwidth. The Efficient use of the available bandwidth is achieved through the transmitter filter, also called the modulating filter. The modulator on the other hand places the signal over a high frequency carrier for efficient transmission. Some of the typical modulation schemes used in digital communication are amplitude shift keying (ASK), frequency shift keying (FSK), pulse amplitude modulation (PAM) and phase shift keying (PSK) modulation. The channel is the medium through which information propagates from the transmitter to the receiver. At the receiver the signal is first demodulated to recover the baseband transmitted signal. The demodulated signal is processed by the receiver filter, also called receiver demodulating filter, which should be ideally matched to the transmitter filter and the channel. Hence physical channel can replace all filters in block diagram. The equalizer in the receiver removes the distortion introduced due to the channel impairments. The decision device provides the estimate of the encoded transmitted signal. The decoder reverses the work of the encoder and removes the encoding effect revealing the transmitted information symbols. This simplified communication system model, while maintaining the basic principle involved, is easy to analyze.

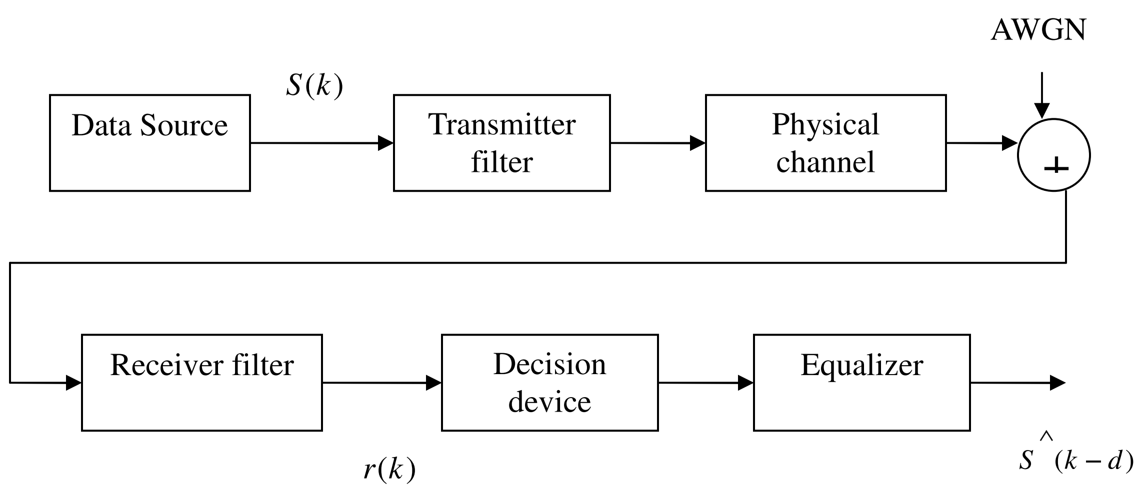


Fig 2.1 Baseband model of digital communication system

2.3 Propagation channel

This section discusses the channel impairments that limit the performance of a DCS. The DCS considered here is shown in Fig 2.1.The transmission of digital pulses over an analogue channel would require infinite bandwidth. An ideal physical propagation channel should behave like an ideal low pass filter represented by its frequency response,

$$H_c(f) = \left|H_c(f)\right| \exp(j\theta f) \tag{2.1}$$

Where $H_c(f)$ represents the Fourier transform (FT) of the channel and θ is the phase response of the channel. The amplitude response of the channel $\left|H_c(f)\right|$ can be defined as,

$$H_c(f) = \begin{cases} k_1 & |f| \leq \omega_c \\ 0 & |f| > \omega_c \end{cases} \tag{2.2}$$

Where k_1 is a constant and ω_c is the upper cutoff frequency. The channel group delay characteristic is given by

$$\tau(f) = -\frac{1}{2\Pi}\frac{d\theta(f)}{df} = k_1 \tag{2.3}$$

Where k_2 is an arbitrary constant. The conditions described in (2.2) and (2.3) constitute fixed amplitude and linear phase characteristics of a channel. This channel can provide distortion free transmission of analogue signal band limited to at least ω_c. Transmission of the infinite bandwidth digital signal over a band limited channel of ω_c will obviously cause distortion. This demands for the infinite bandwidth digital signal is band limited to at least at least ω_c, to guarantee distortion free transmission. This work is done with the aid of transmitter and receiver filters shown in Fig 2.1. The combined frequency response of the physical channel, transmitter filter and the receiver filter can be represented as,

$$H(f) = H_T(f) H_c(f) H_R(f) \tag{2.4}$$

Where $H_T(f), H_c(f), H_R(f)$ represents the FT of the transmitter, channel and receiver respectively. When the receiver filter is matched to the combined response of the propagation channel and the transmitter filter, the system provides optimum signal to noise ratio (SNR) at the sampling instant. As channel impulse response is not known beforehand, hence the receiver filter impulse response $h_R(t)$ is generally matched to the transmitter filter impulse response $h_T(t)$ For this condition to be satisfied the frequency response of both the transmitter and receiver filters must be complex conjugate to each other. For the ideal channel case though it is very difficult but is possible by using a raised cosine filter.

2.3.1 Inter symbol interference (ISI)

The cascade of the transmitter filter $h_T(t)$ the channel $hc(t)$ and the receiver $h_R(t)$ matched filter and the T spaced sampler in the communication system shown in Fig 2.1 can be modeled by a digital FIR filter. The noise at the equalizer input is correlated due to the presence of the matched filter. To take care of this, and since it is easier to deal with a white noise sequence in the equalizer, the equalizer is generally preceded by a noise whitening filter. This combined channel due to the transmitter filter, propagation channel, receiver filter, noise whitening filter and the T spaced sampler can be modeled by the digital FIR filter represented in Fig 2.2. Here the channel observed output $r(k)$ is given by the sum of the noise free channel output $\hat{r}(k)$, Which in turn is formed by the convolution of the transmitted sequence $s(k)$ with the channel taps a_i , $0 \leq i \leq n_c - 1$ and AWGN $\eta(k)$.

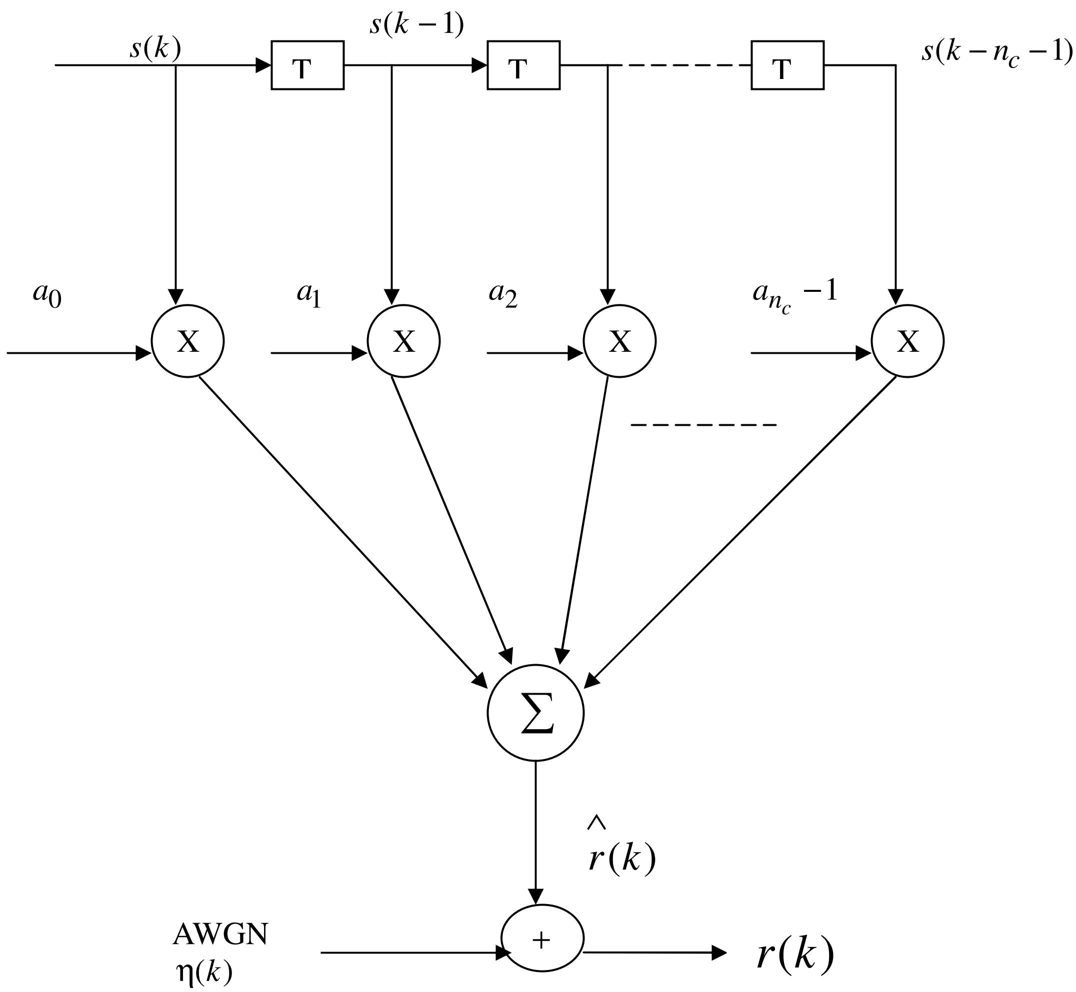


Fig.2.2 Finite impulse response channel model

The channel impulse response in the z -domain can be represented by the equation

$$H(z) = \sum_{i=0}^{n_c-1} a_i z^{-i} \tag{2.5}$$

Where, the channel provides dispersion up to $n_c - 1$ samples. This *discrete time white noise linear filter model* of the continuous channel will be used in the remaining part of the thesis for evaluation of equalizer algorithms. Here the AWGN, $\eta(k)$ is characterized by its variance σ_n^2.

2.3.2 Co-channel interference (CCI) and adjacent channel interference (ACI)

CCI and ACI occur in communication systems due to multiple access techniques using space, frequency or time. When the signal of interest in a communication system is corrupted by another signal occupying the same frequency band, CCI occurs. However, the source of ACI can be attributed to inadequate inter carrier spacing and non ideal receiver filter characteristics. In twisted pair cables CCI occurs due to interference of signals between different twisted pairs and is termed near end cross talk (NEXT), and far end cross talk (FEXT)[27,28]. In DCR the CCI can be attributed to interference from cells of neighboring clusters using the same carrier frequency[29] and ACI is due to inter carrier spacing between different cells in time division multiple access (TDMA)[30] and inter carrier spacing among carriers in the same cell in FDMA[29,31,32] systems. The frequency spectrum of the signals that carry the desired signal, the CCI and ACI signals is presented in Fig 2.3

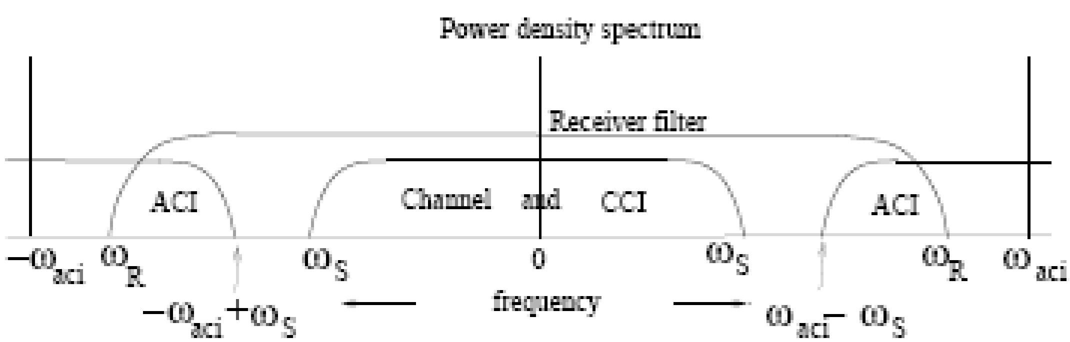


Fig.2.3. Spectrum of desired signal, CCI and ACI in DCS [36]

Here the signal of interest occupies a double sided bandwidth of ω_s The CCI signal also occupies the same frequency band. The ACI signal centre frequency is spaced at ω_{aci}

with respect to the desired carrier. The receiver filter rejects signal beyond ω_R . The guard band provided in the system is $\omega_{aci_s} - 2\omega_s$. From the Fig 2.3 it can be seen that a portion of the signal spectrum in the neighboring carrier with respect to the signal of interest is received by the receiver filter and this signal is the main cause of ACI. The main reasons for this ACI can be attributed to non ideal cutoff characteristics of the receiver filter and close spacing of the carrier frequencies. Discrete time representation of the channel, the co-channel and the adjacent channel interferers using digital filters is presented in Fig 2.4. This system consists of a channel $H(z)$ corrupted with L, CCI sources $H_{co\_j}(z)$, $1 \le j \le L$ and B, ACI sources $H_{aci\_j}(z)$, $1 \le j \le B$ each of which can be represented in the form of a FIR filter of the type presented in Fig 2.2. The channel is also additionally corrupted with AWGN, $\eta(k)$. The total CCI and ACI are presented as $\hat{r}_{co}(k)$ and $\hat{r}_{ac}(k)$ respectively. Here $s_0(k)$ are the transmitted symbols from the desired channel, $s_i(k)$ $1 \le i \le L$ represent the transmitted symbols from the co-channel i and $S_{aci\_j}(k)$ represent the transmitted symbols from adjacent channel j .

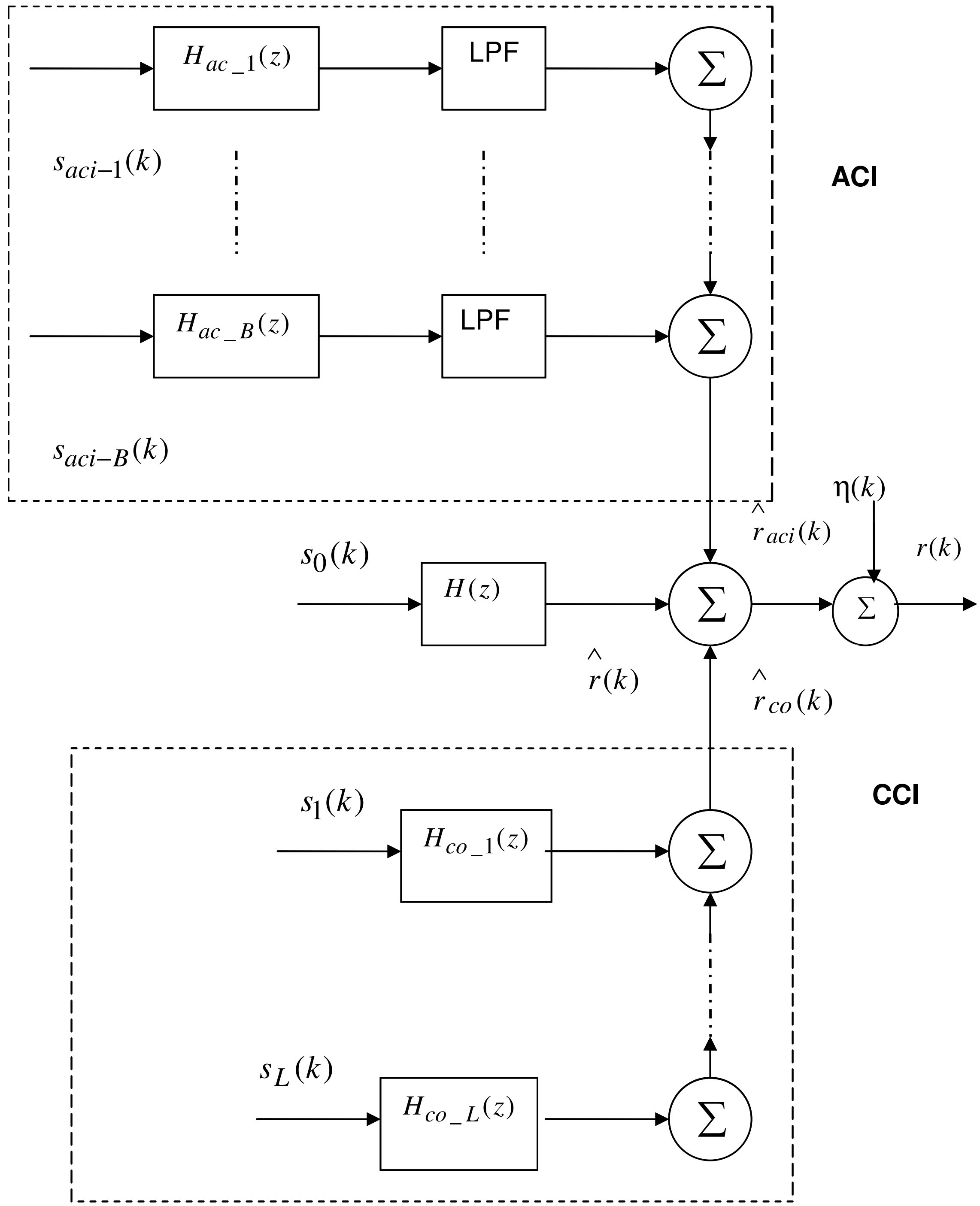


Fig.2.4. *FIR filter implementation of channel, CCI and ACI in digital communication system*

2.4 Equalizer classification

This section provides adaptive equalizer classification and specifies the domain of the investigation undertaken in this thesis. The general equalizer classification is presented in Fig 2.5. In general the family of adaptive equalizers can be classified as *supervised equalizers* and *unsupervised equalizers*. The channel distortions introduced into the transmitted signal in the process of transmission can be conveniently removed by transmitting a *training signal* or *pilot signal* periodically during the transmission of information. A replica of this pilot signal is available at the receiver and the receiver uses this to update its parameters during the training period. These kinds of equalizers are known as supervised equalizers. However, the constraints associated with communication systems like digital television and digital radio do not provide the scope for the use of a training signal. In this situation the equalizer needs some form of unsupervised or *self recovery* method to update its parameters so as to provide near optimal performance. These equalizers are called *blind equalizers*. After training, the equalizer is switched to *decision directed* mode, where the equalizer can update its parameters based on the past detected samples. This thesis investigates supervised equalizers in general.

The process of supervised equalization can be achieved in two forms. These are *sequence estimation* and *symbol-by-symbol estimation*. The sequence estimator uses the sequence of past received samples to estimate the transmitted symbol. For this reason this forms of equalizer is considered as an infinite memory equalizer and is termed MLSE. The MLSE can be implemented with the Viterbi Algorithm]. An infinite memory sequence estimator provides the best bit error ratio (BER) performance for equalization of time invariant channels. The symbol-by-symbol equalizer on the other hand works as a finite memory equalizer and uses a fixed number of input samples to detect the transmitted symbol. The optimum decision function for this type of equalizer is given by MAP criterion and can be derived by Bayes's theory[33]. Hence this optimum finite memory equalizer is

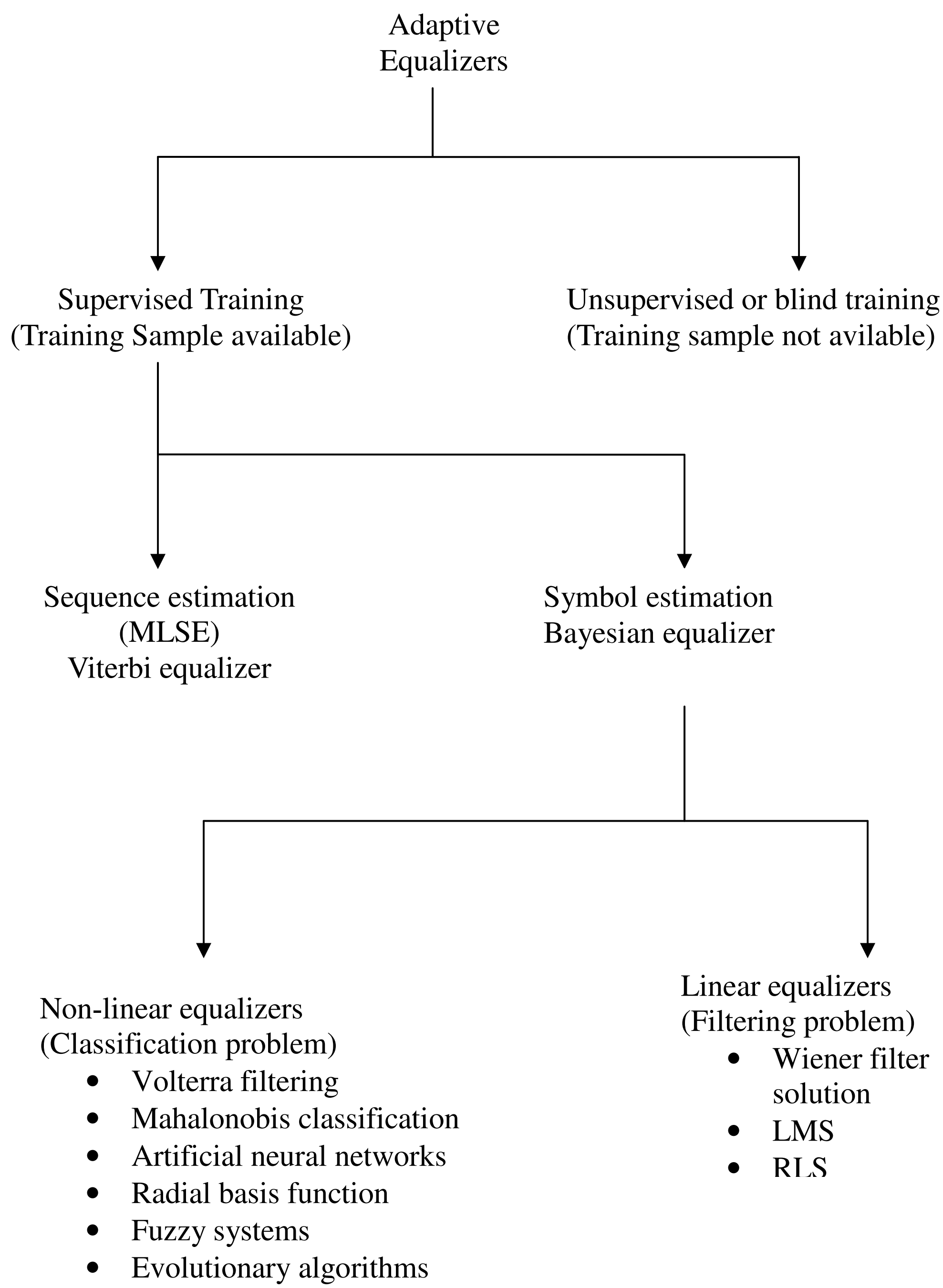


Fig.2.5. Classification of adaptive equalizers

also called the Bayesian equalizer[34]. An infinite memory Bayesian equalizer can provide a performance better than the MLSE, but its computational complexity is very large. A finite memory Bayesian equalizer can provide performance comparable to the MLSE but with a reduced computational complexity.

The Bayesian equalizer provides the lower performance bound for symbol-by-symbol equalizers in terms of probability of error or BER and can be implemented with *linear* or *nonlinear* systems. The linear adaptive equalizer is a linear FIR adaptive filter trained with an adaptive algorithm like the LMS, RLS or lattice algorithm. These linear equalizers treat equalization as inverse filtering and during the process of training optimize a certain performance criteria. Like minimum mean square error (MMSE) or amplitude distortion. Linear equalizers trained with MMSE criterion provide the Wiener filter solution. Recent advances in nonlinear signal processing techniques have provided a rich variety of nonlinear equalizers. Some of the equalizers developed with these processing techniques are based on Volterra filters, ANN, perceptions, MLP, RBF networks, fuzzy filters and fuzzy basis functions. All of these nonlinear equalizers, during their training period, optimize some form of a cost function like the MSE or probability of error and have the capability of providing the optimum Bayesian equalizer performance in terms of BER. The nonlinear equalizers treat equalization as a pattern classification process where the equalizer attempts to classify the input vector into a number of transmitted symbols. The bacterial foraging based equalizers investigated in this thesis falls in this category.

Another form of nonlinear equalizer that can be constructed with any of the symbol-by-symbol based equalizers is the DFE, where previously made decisions are used for estimating the present and the future decisions. This equalizer is also considered as an infinite memory equalizer. The conventional DFE using a linear filter is designated as a nonlinear equalizer in wide Verities of communication literature since the decision function used here forms a nonlinear combination of the received samples which is, in fact the linear combination of the received samples and previously detected samples. In this thesis the term nonlinear equalizers is used exclusively for those equalizers that provide a nonlinear decision function based on received samples or the received samples along with previously detected samples.

2.5 Optimal symbol-by-symbol equalizer: Bayesian equalizer

The discrete time model of a digital communication system is as shown in Fig.2.6.

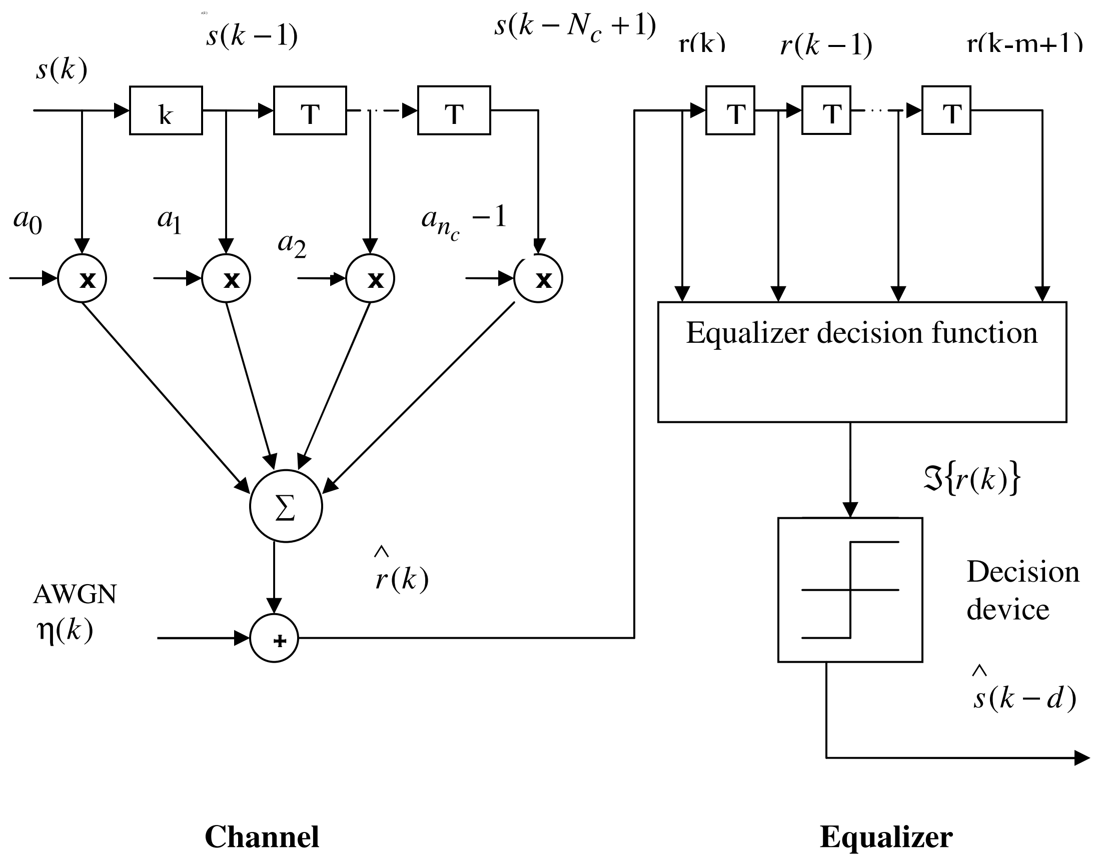


Fig.2.6. Discrete time model of digital communication system

The equalizer uses an input vector $r(k) \in \Re^m$ the m dimensional space. The term m is the equalizer length and the equalizer order can be considered as $m-1$.The equalizer provides a decision function $\Im\{r(k)\}$ based on the input vector and this is passed through a decision device to provide the estimate of transmitted signal $\hat{s}(k-d)$ where d is a delay associated with equalizer decision. The communication system is assumed to be a two level PAM system where the transmitted sequence is drawn $s(k)$ from a independent identically distributed (I.I.d) sequence comprising of $\{\pm 1\}$ symbols. The noise source $\eta(k)$ is assumed to be zero mean white Gaussian with a variance of σ_n^2. The received signal r(k) at the sampling instant 2 can be represented as,

$$
\begin{aligned}
r(k) &= \hat{r}(k) + \eta(k) \\
&= \sum_{i=0}^{n_c - 1} a_i s(k-i) + \eta(k)
\end{aligned}
\tag{2.6}
$$

2.6 Symbol-by-symbol linear equalizers

This section introduces the concept of the linear equalizer. As discussed in section 2.4, the linear equalizers in this thesis refer to equalizers that provide a decision based on the linear combination of the input to the equalizer. If decision feedback is employed, the linear equalizer provides a decision function based on the linear combination of received samples and previously detected samples. The structure of a linear equalizer is presented in Fig 2.6. The equalizer consists of a T spaced tapped delay line (TDL) which receives the receiver sampled input vector $r(k) = [r(k), r(k-1).....r(k-m)]^T$ and provides an output $y(k)$ by weighted sum computation of input vector $r(k)$ with weight vector w. The output is computed once per symbol and can be represented as

$$
y(k) = \sum_{i=0}^{m-1} w_i r(k-i) \tag{2.7}
$$

The weight vector w optimizes one of the performance criteria like zero forcing (ZF) or MMSE criteria. The decision device presented at the output of the filter provides the transmitted signal constellation.

The ZF criterion is defined as the worst case ISI at the output of the equalizer. The condition for minimization of peak distortion can be presented as

$$
C(Z) = \frac{1}{H(Z)} \tag{2.8}
$$

Here $C(Z)$ is the equalizer impulse response. With this, the combined equalizer and the channel response is zero for all but one coefficient. From the equalizer condition presented in (2.7) it can be seen that, for FIR channels, the equalizer is realizable when the zeros of the channel are inside the unit circle in the Z-plane. When the zeros are outside the unit circle, the equalizer becomes unstable and hence unrealizable. Equalization of this type of channel can be overcome by the introduction of a nonzero decision delay 0.

The MMSE criteria provide equalizer tap coefficients $w(k)$ to minimize the mean square error at the equalizer output before the decision device. This condition can be represented as

$$J = \varepsilon|e(k)|^2 \tag{2.9}$$

$$e(k) = s(k-d) - y(k) \tag{2.10}$$

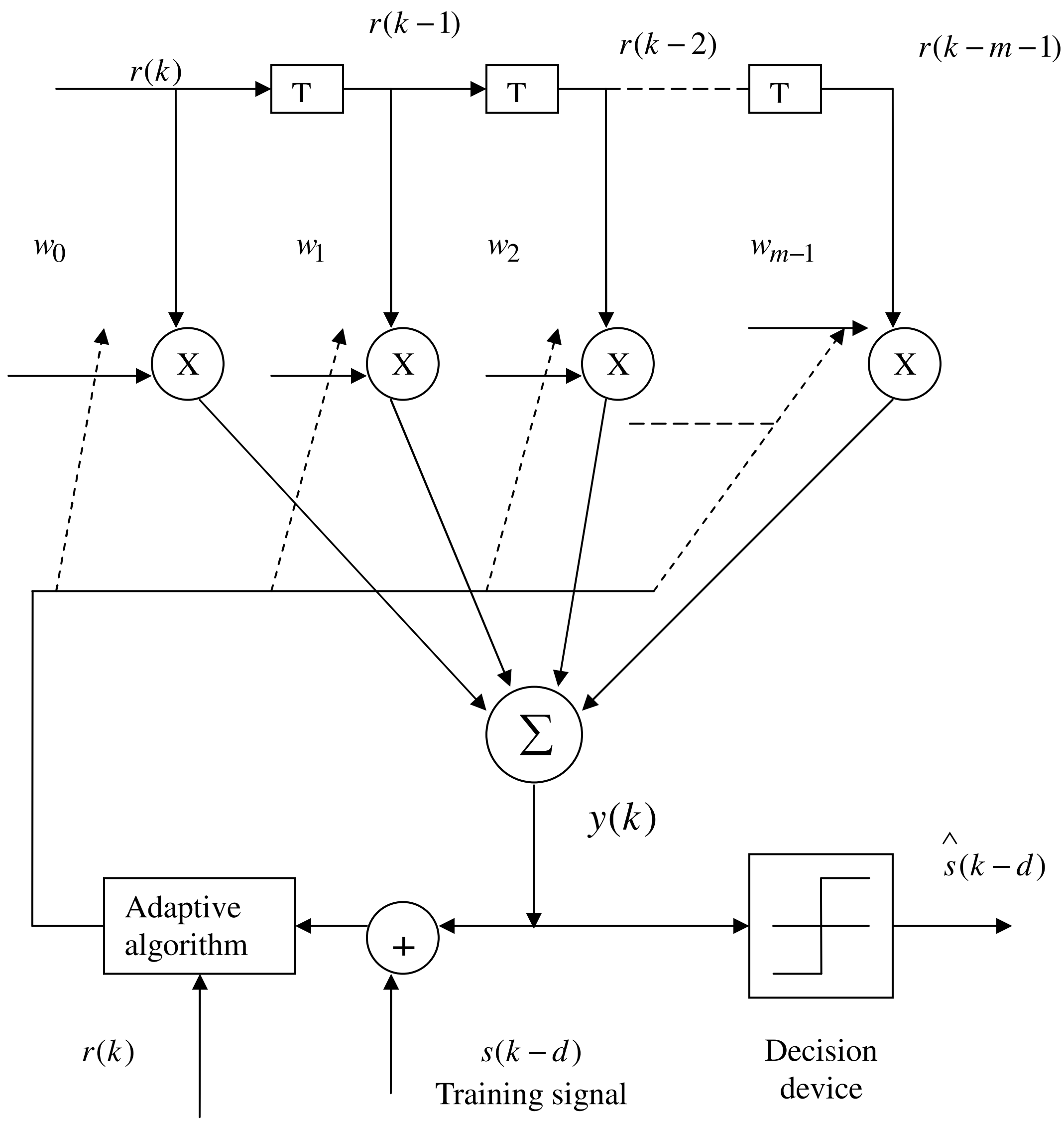


Fig.2.7 Structure of a linear equalizer

Where $e(k)$ is the error associated with filter output $y(k)$. The equalizer designed using ZF criteria neglects the effect of noise. However, the MMSE criteria optimize the equalizer weights for minimizing the MMSE under noise and ISI. Minimization of MMSE criteria provides equalizers that satisfy the Wiener criterion]. The evaluation of the equalizer weights with these criteria requires computation of matrix inversion and the knowledge of the channel, which in most cases is not available. However, adaptive algorithms like LMS] and RLS can be used to recursively update the equalizer weights during the training period. The convergence properties and the performance of linear equalization techniques have been well documented in the literature.

2.7 Symbol-by-symbol adaptive nonlinear equalizers

Some of the popular forms of nonlinear equalizers are introduced in this section. Nonlinear equalizers treat equalization as a nonlinear pattern classification problem and provide a decision function that partitions the input space $\left|\Re^m\right|$ to the number of transmitted symbols. As a result the equalizer assigns the input vector to one of the signal constellations. The nonlinear equalizers introduced in this section are based on the RBF networks. Some of the other Forms of nonlinear equalizers based on the recurrent RBF, ANN, the recurrent ANN, the Volterra filters, the functional link networks and Mahalobonis classifiers have not been discussed.

2.7.1 Radial basis function equalizer

The RBF network was originally developed for interpolation in multidimensional space. The schematic of this RBF network with m inputs and a scalar output is presented in Fig 2.7. This network can implement a mapping $f_{rbf} : \Re^m \to \Re$ by the function,

$$f_{rbf}\{x(k)\} = \sum w_i \phi(\|x(k)\| - \rho_i) \tag{2.11}$$

Where $x(k) \in \Re^m$ is the input vector, $\phi(.)$ is the given function from $\Re^+$ to $\Re$, $w_i, 1 \le i \le N_r$ are weights and $\rho_i \in \Re^m$ are known as RBF centers. This RBF structure can be extended for multidimensional output as well.

Possible choices for the radial basis function $\phi(\gamma)$ include a thin plate spline,

$$\Phi(\gamma) = \frac{\gamma}{\sigma_r^2} \log(\frac{r}{\sigma_r}) \tag{2.12}$$

A multi quadratic,

$$\phi(\gamma) = \sqrt{(\gamma^2 + \sigma_r^2)} \tag{2.13}$$

An inverse multi quadratic,

$$\phi(\gamma) = \frac{1}{\sqrt{(\gamma^2 + \sigma_r^2)}} \tag{2.14}$$

and Gaussian kernel,

$$\phi(\gamma) = \exp(\frac{-\gamma^2}{2\sigma^2}) \tag{2.15}$$

Here, the parameter σ^2 controls the radius of influence of each basis functions and determines how rapidly the function approaches 0 with γ . The Gaussian and the inverse multi-quadratic kernel provide bounded and localized properties such that $\phi(\gamma) \to 0$ as $\gamma \to \infty$. Broomhead and Lowe reinterpreted the RBF network as a least square estimator which led to its wide spread use in signal processing applications such as time series prediction system identification, interference cancellation, radar signal processing, pattern classification and channel equalization. In signal processing applications the RBF inputs are presented through a TDL. Training of the RBF networks involves setting the parameters for the centers ρ_i , spread σ_r and the linear weights w_i .

The RBF networks are easy to train since the training of centers, spread parameter and the weights can be done sequentially and the network offers a nonlinear mapping, maintaining its linearity in parameter structure at the output layer. One of the most popular schemes employed for training the RBF in a supervised manner is to estimate the centers using a clustering algorithm like the k -means clustering and setting σ_r^2 to an estimate of input noise variance calculated from the centre estimation error. The output layer weights can be trained using popular stochastic gradient LMS algorithm. Other schemes for RBF training involve selecting a large number of candidate centers

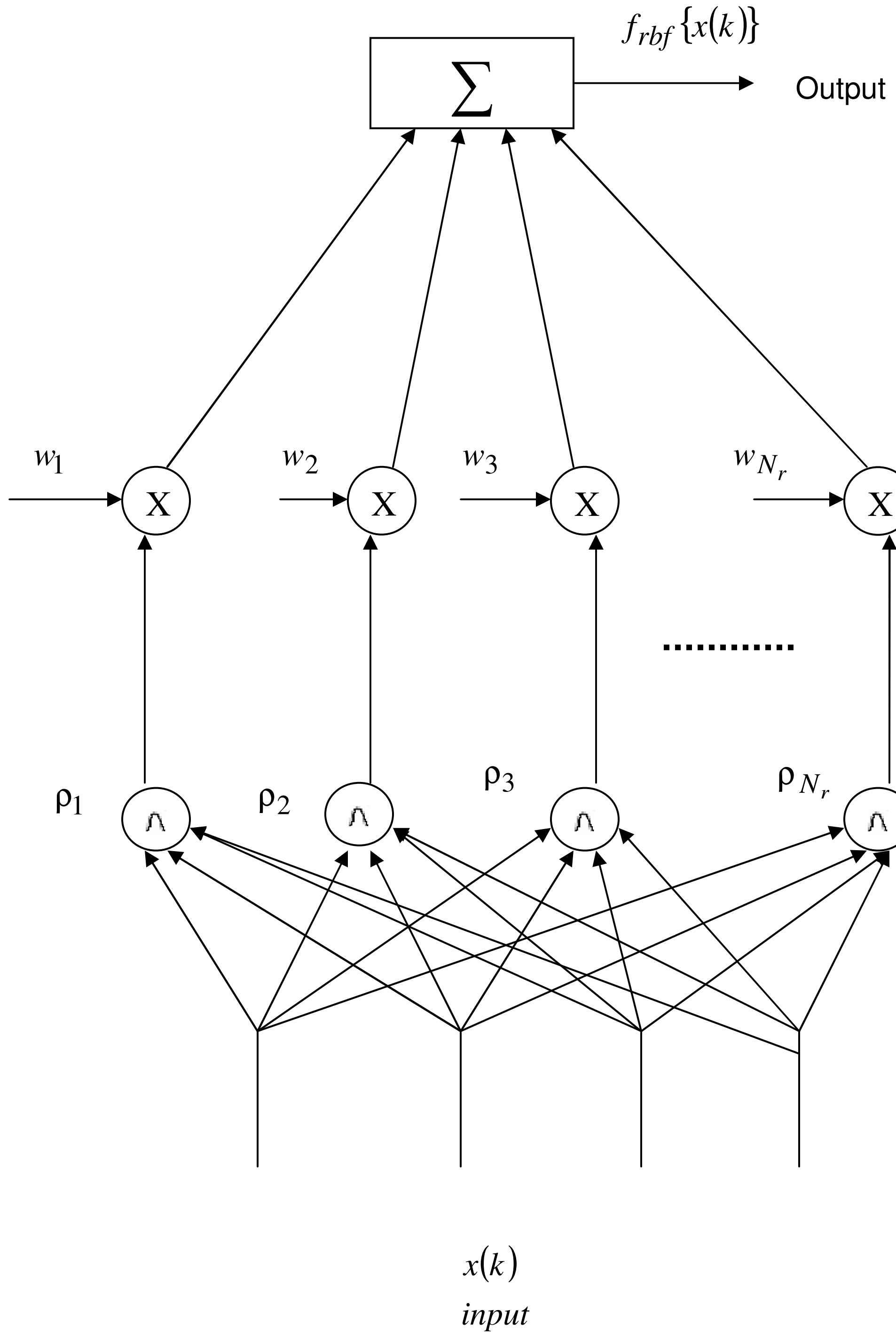


Fig.2.8. A radial basis function network for signal processing applications

initially and use the orthogonal least squares (OLS) algorithm to pick a subset of the centers that provides near optimal performance. The MLP back propagation algorithm can also be used to train the RBF centers. In early RBF equalizers the RBF centers were selected at random, picked from a few of the initial input vectors. The weights were updated using supervised training by the LMS algorithm or its momentum version. This resulted in equalizers with large number of centers making the network computationally complex. Chen proposed the OLS algorithm for selecting an optimum number of centers from a large number of candidate centers, resulting in near optimal performance. Subsequently, the close relationship between the RBF network and the Bayesian equalizer was found and this provided the parametric implementation of the Bayesian equalizers with the RBF. In these equalizers supervised k -means clustering provides the estimate of the centers while linear weights are estimated using the LMS algorithm. With the development of RBFs that could handle complex signals, they were used for equalization in communication systems with complex signal constellation. Cha proposed the stochastic gradient algorithm to adapt all the RBF parameters and used this technique to equalize 4-QAM digital communication systems.

The RBF equalizer can provide optimal performance with small training sequences but they suffer from computational complexity. The number of RBF centers required in the equalizer increases exponentially with equalizer order and the channel delay dispersion order. This increases all the computations exponentially. In a varied implementation the RBF with scalar centers results in a reduction of computational complexity. The issues relating to the RBF equalizer design have been discussed extensively in [25].

2.8 Conclusion

In this chapter the optimum symbol-by-symbol equalizer decision function was derived and its implementation using the RBF was presented. The RBF equalizer using bacterial foraging algorithm is discussed in chapter 4.

Chapter-3

Evolutionary Algorithms

3.1 Introduction

Evolutionary algorithms are stochastic search methods that mimic the metaphor of natural biological evolution. Evolutionary algorithms operate on a population of potential solutions applying the principle of survival of the fittest to produce better and better approximations to a solution. At each generation, a new set of approximations is created by the process of selecting individuals according to their level of fitness in the problem domain and breeding them together using operators borrowed from natural genetics. This process leads to the evolution of populations of individuals that are better suited to their environment than the individuals that they were created from, just as in natural adaptation.

This chapter is organized as follows. Following this introduction section 3.2 describes the working of evolutionary algorithms, section 3.3 gives some examples of evolutionary algorithms, section 3.4 presents the basic idea of bacterial foraging optimization, section 3.5 discusses the bacterial foraging algorithm, section 3.6 presents the algorithms flowchart, section 3.7 discusses the algorithm parameter choices, finally section 3.8 provides the concluding remarks.

3.2 working of 'Evolutionary Algorithms'

Evolutionary algorithms model natural processes, such as selection, recombination, mutation, migration, locality and neighborhood. Fig.3.1 shows the structure of a simple evolutionary algorithm. Evolutionary algorithms work on populations of individuals instead of single solutions. In this way the search is performed in a parallel manner.

At the beginning of the computation a number of individuals (the population) are randomly initialized. The objective function is then evaluated for these individuals. The first/initial generation is produced.

If the optimization criteria are not met the creation of a new generation starts. Individuals are selected according to their fitness for the production of offspring. Parents are recombined to produce offspring. All offspring will be mutated with a certain probability. The fitness of the offspring is then computed. The offspring are inserted into the population replacing the parents, producing a new generation. This cycle is performed until the optimization criteria are reached.

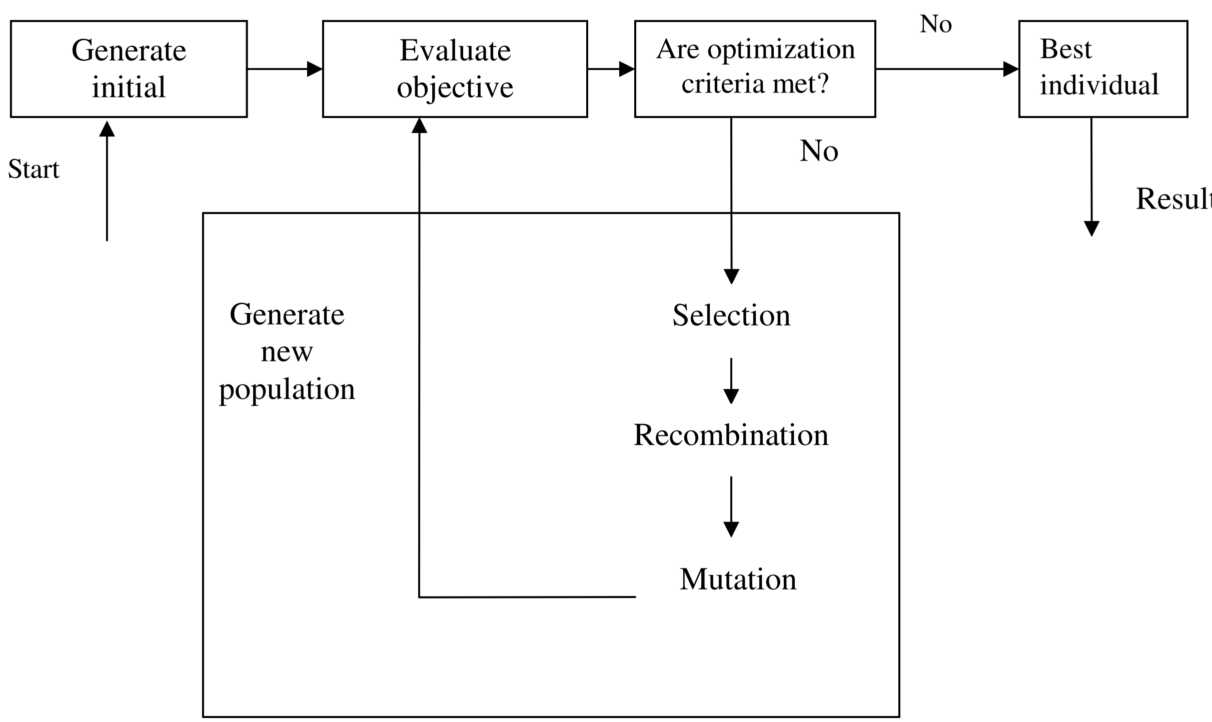


Fig 3.1 Structure of a single population evolutionary algorithm

Such a single population evolutionary algorithm is powerful and performs well on a wide variety of problems. However, better results can be obtained by introducing multiple subpopulations. Every subpopulation evolves over a few generations isolated (like the single population evolutionary algorithm) before one or more individuals are exchanged between the subpopulation. The multi-population evolutionary algorithm models the evolution of a species in a way more similar to nature than the single population evolutionary algorithm. Fig.3.2. shows the structure of such an extended multi-population evolutionary algorithm.

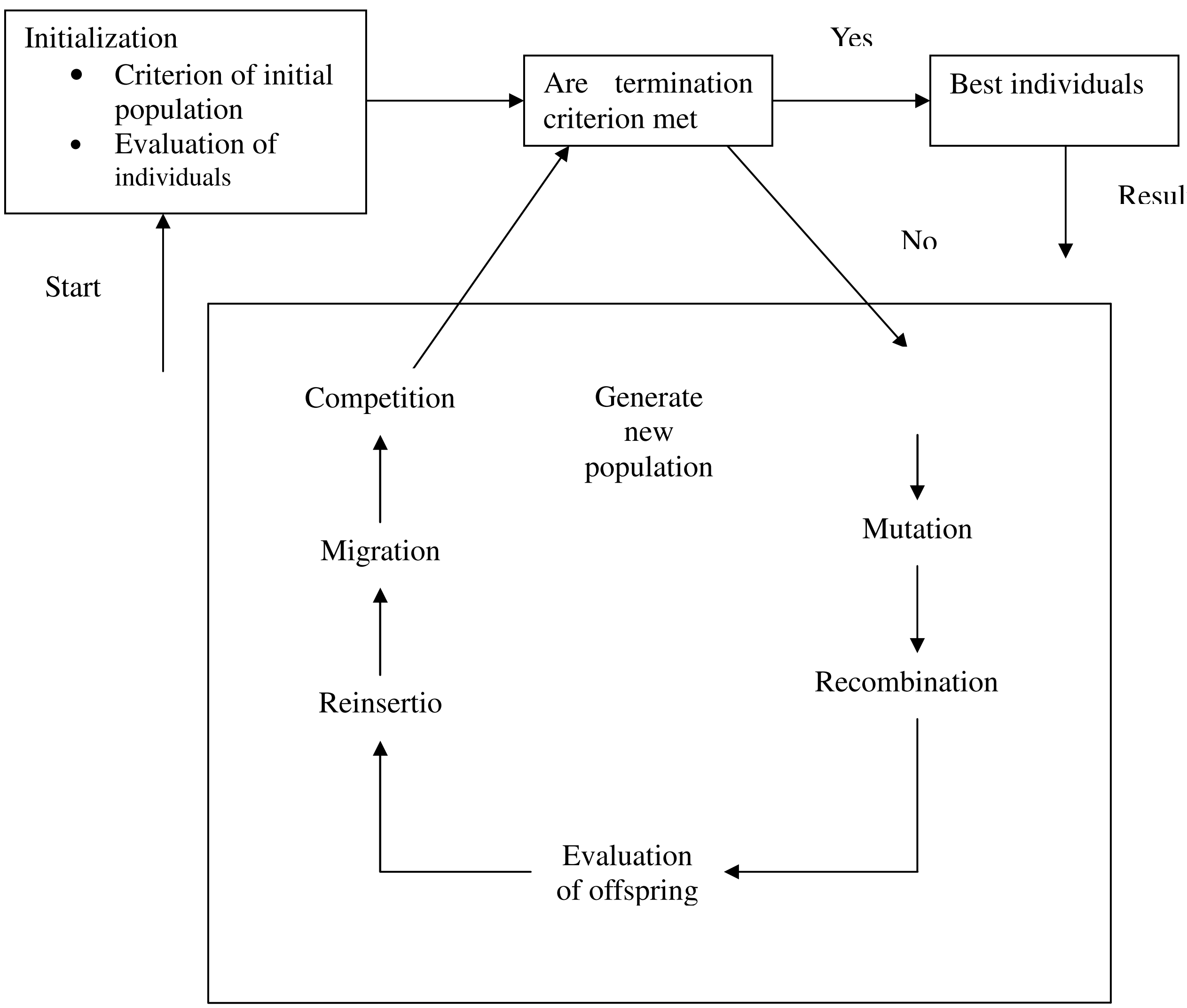


Fig 3.2 Structure of an extended multipopulation evolutionary algorithm

From the above discussion, it can be seen that evolutionary algorithms differ substantially from more traditional search and optimization methods. The most significant differences are:

- Evolutionary algorithms search a population of points in parallel, not just a single point.
- Evolutionary algorithms do not require derivative information or other auxiliary knowledge; only the objective function and corresponding fitness levels influence the directions of search.
- Evolutionary algorithms use probabilistic transition rules, not deterministic ones.
- Evolutionary algorithms are generally more straightforward to apply, because no restrictions for the definition of the objective function exist.
- Evolutionary algorithms can provide a number of potential solutions to a given problem. The final choice is left to the user. (Thus, in cases where the particular problem does not have one individual solution, for example a family of pareto-optimal solutions, as in the case of multi-objective optimization and scheduling problems, then the evolutionary algorithm is potentially useful for identifying these alternative solutions simultaneously.)

3.3 Some examples of EA

Genetic algorithm - This is the most popular type of EA. One seeks the solution of a problem in the form of strings of numbers (traditionally binary, although the best representations are usually those that reflect something about the problem being solved - these are not normally binary), virtually always applying recombination operators in addition to selection and mutation.

Evolutionary programming - Like genetic programming, only the structure of the program is fixed and its numerical parameters are allowed to evolve.

Evolution strategy - Works with vectors of real numbers as representations of solutions, and typically uses self-adaptive mutation rates.

Genetic programming - Here the solutions are in the form of computer programs, and their fitness is determined by their ability to solve a computational problem.

Learning classifier system - Instead of a using fitness function, rule utility is decided by a reinforcement learning technique.

3.2.1 Related techniques

Differential evolution - Based on vector differences and is therefore primarily suited for numerical optimization problems.

Particle swarm optimization - Based on the ideas of animal flocking behavior. Also primarily suited for numerical optimization problems.

Ant colony optimization - Based on the ideas of ant foraging by pheromone communication to form path. Primarily suited for combinatorial optimization problems.

Bacterial foraging - Based on the ideas of bacteria foraging by swimming and tumbling. Primarily suited for combinatorial optimization problems.

3.4 Basic Bacterial Foraging Optimization

Natural selection tends to eliminate animals with poor foraging strategies and favor the propagation of genes of those animals that have successful foraging strategies, since they are more likely to enjoy reproductive success. After many generations, poor foraging strategies are either eliminated or shaped into good ones. This activity of foraging led the researchers to use it as optimization process. The *E. coli* bacteria that are present in our intestines also undergo a foraging strategy. The control system of these bacteria that dictates how foraging

should proceed can be subdivided into four sections, namely, chemotaxis, swarming, reproduction, and elimination and dispersal.

3.4.1 Chemotaxis: This process in the control system is achieved through swimming and tumbling via Flagella. Each flagellum is a left-handed helix configured so that as the base of the flagellum (i.e., where it is connected to the cell) rotates counterclockwise, as viewed from the free end of the flagellum looking toward the cell, it produces a force against the bacterium so it pushes the cell. On the other hand, if they rotate clockwise, each flagellum pulls on the cell, and the net effect is that each flagellum operates relatively independently of others, and so the bacterium tumbles about. Therefore, an *E. coli* bacterium can move in two different ways; it can run (swim for a period of time) or it can tumble, and alternate between these two modes of operation in the entire lifetime. To represent a tumble, a unit length random direction, say $\phi(j)$, is generated; this will be used to define the direction of movement after a tumble. In particular

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + c(i)\phi(j) \tag{3.1}$$

Where θ^i (j+1, k, l) represents the i th bacterium at j th chemotactic k th reproductive and l th elimination and dispersal step. C(i) is the size of the step taken in the random direction specified by the tumble (run length unit).

3.4.2 Swarming: When a group of *E. coli* cells is placed in the center of a semisolid agar with a single nutrient chemo-effecter (sensor), they move out from the center in a traveling ring of cells by moving up the nutrient gradient created by consumption of the nutrient by the group. Moreover, if high levels of succinate are used as the nutrient, then the cells release the attractant aspartate so that they congregate into groups and, hence, move as concentric patterns of groups with high bacterial density. The spatial order results from outward movement of the ring and the local releases of the attractant; the cells provide an attraction signal to each other so they swarm together. The mathematical representation for swarming can be represented by

$$\begin{aligned} J_{cc}(\theta, P(j,k,l)) &= \sum_{i=1}^{S} J_{cc}^{i}(\theta, \theta^i(j,k,l)) \\ &= \sum_{i=1}^{S} [-d_{attrac\tan t} \exp(-w_{attrac\tan t} \sum_{m=1}^{p} (\theta_m - \theta_m^i)^2)] \\ &+ \sum_{i=1}^{S} [h_{repellent} \exp(-w_{attrac\tan t} \sum_{m=1}^{p} (\theta_m - \theta_m^i)^2)] \end{aligned} \tag{3.2}$$

where $J_{cc}(\theta, P(j,k,l))$ is the cost function value to be added to the actual cost function to be minimized to present a time varying cost function, S is the total number of bacteria, P is the number of parameters to be optimized which are present in each bacterium, and $d_{attarct}, w_{attract}, h_{repellent}, w_{repellent}$ are different coefficients that are to be chosen properly.

3.4.3 Reproduction: The least healthy bacteria die and the other healthier bacteria each split into two bacteria, which are placed in the same location. This makes the population of bacteria constant.

3.4.4 Elimination and Dispersal: It is possible that in the local environment, the lives of a population of bacteria changes either gradually (e.g., via consumption of nutrients) or suddenly due to some other influence. Events can occur such that all the bacteria in a region are killed or a group is dispersed into a new part of the environment. They have the effect of possibly destroying the chemotactic progress, but they also have the effect of assisting in chemotaxis, since dispersal may place bacteria near good food sources. From a broad perspective, elimination and dispersal are parts of the population-level long-distance motile behavior. This section is based on the work in[35]. As this paper concentrates in applying the new method to harmonic estimation, the in depth discussion over the bacterial foraging strategy is not dealt. The detailed mathematical derivations, as well as the theoretical aspect of this new concept are presented in[35].

3.5 Bacterial foraging - Algorithm

For initialization, we must choose P , S , N_c , N_s , N_{re} , N_{ed} , P_{ed} , and the C(i),i=1,2…… S. In case of swarming, we will also have to pick the parameters of the cell-to-cell attractant functions; here we will use the parameters given above. Also, initial values for the θ^i, i = 12….S must be chosen. Choosing these to be in areas where an optimum value is likely to exist is a good choice. Alternatively, we may want to simply randomly distribute them across the domain of the optimization problem. The algorithm that models bacterial population chemotaxis, swarming, reproduction, elimination, and dispersal is given here (initially j=k=l=0). For the algorithm, note that updates to the θ^i

automatically result in updates to P. Clearly, we could have added a more sophisticated termination test than simply specifying a maximum number of iterations.

1) *Elimination-dispersal loop*: l=l+1

2) *Reproduction loop*: k=k+1

3) *Chemotaxis loop:* j=j+1

a) For i =12….S, take a chemotactic step for bacterium i as follows.

b) Compute $J(i,j,k,l)$ Let $J(i,j,k,l) = J(i,j+1,k,l) + J_{cc}(\theta^{i}(j+1,k,l).P(j+1,k,l)$

(i.e., add on the cell-to-cell attractant effect to the nutrient concentration).

c) Let $J_{last} = J(i,j,k,l)$ to save this value since we may find a better cost via a run.

d) *Tumble*: Generate a random vector $\Delta(i) \in \Re^{p}$ with each element $\Delta_{m}(i)$,m=1,2…..p a random number on [-1,1].

e) *Move*: Let $\theta^{i}(j+1,k,l) = \theta^{i}(j+1,k,l) + C(i)\ \Delta(i)\dfrac{1}{\sqrt{\Delta^{t}(i)\Delta(i)}}$

This results in a step of size $C(i)$ in the direction of the tumble for bacterium i.

f) Compute $J(i,j+1,k,l)$ and then let

$$J(i,j+1,k,l) = J(i,j+1,k,l) + J_{cc}(\theta^{i}(j+1,k,l).P(j+1,k,l)$$

g) *Swim* (note that we use an approximation since we decide swimming behavior of each cell as if the bacteria numbered {1, 2…i} have moved and {i+1, i+2…s} have not; this is much simpler to simulate than simultaneous decisions about swimming and tumbling by all bacteria at the same time):

I. Let m = 0 (counter for swim length).

II. While m < N_s (if have not climbed down too long)

• Let m = m + 1

• I f $J(i, j+1, k, l) < J_{last}$ (if doing better),

let $J_{last} = J(i, j+1, k, l)$ and let $\theta^i(j+1, k, l) = \theta^i(j, k, l) +$ c(i) $\Delta(i)\dfrac{1}{\sqrt{\Delta^t(i)\Delta(i)}}$ and use this $\theta^i(j+1, k, l)$ to compute the new $J(i, j+1, k, l)$ as we did in f).

• Else, let m=N_s this is the end of the while statement.

h) Go to next bacterium $(i+1)$ if $i \neq S$ (i.e., go to b) to process the next bacterium).

4) If $j < N_c$, go to step 3. In this case, continue chemotaxis, since the life of the bacteria is not over.

5) *Reproduction*: a) For the given k and l, and for each i =1, 2…..S let J^i_{health} = $\sum_{j=1}^{N_c+1} J(i, j, k, l)$ be the health of bacterium i (a measure of how many nutrients it got over its lifetime and how successful it was at avoiding noxious substances). Sort bacteria and chemotactic parameters $C(i)$ in order of ascending cost J_{health} (higher cost means lower health).

b) The S_r bacteria with the highest J_{health} values die and the other S_r bacteria with the best values split (and the copies that are made are placed at the same location as their parent).

6) If K<N_{re} go to step 2. In this case, we have not reached the number of specified reproduction steps, so we start the next generation in the chemotactic loop.

7) *Elimination-dispersal*: For $i = 1, 2 \ldots S$ .With probability P_{ed} , eliminate and disperse each bacterium (this keeps the number of bacteria in the population constant). To do this, if you eliminate a bacterium, simply disperse one to a random location on the optimization domain.

8) If $l < N_{ed}$ then go to step 1; otherwise end.

3.6 Algorithm flowchart

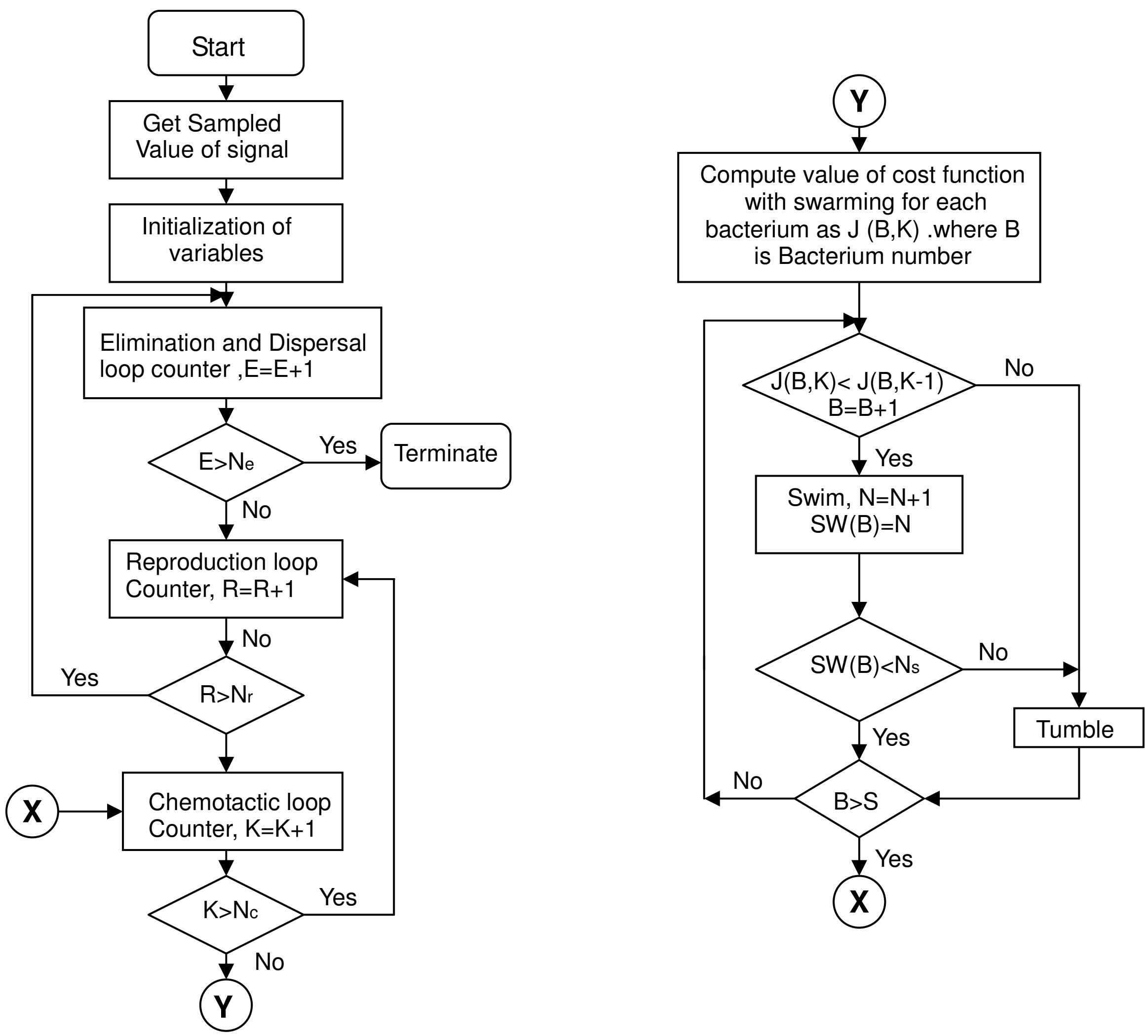


Fig.3.3Fflow chart of the algorithm

3.7 Guidelines for Algorithm Parameter Choices

3.7.1 Size of population 'S': Increasing the size of *S* can significantly increase the computational complexity of the algorithm. However, for larger values of *S*, it is more likely at least some bacteria near an optimum point should be started, and over time, it is then more likely that many bacterium will be in that region, due to either chemotaxis or reproduction.

3.7.2 Length of chemotactic step 'C(i)': If the C(i)values are too large, then if the optimum value lies in a valley with steep edges, the search will tend to jump out of the valley, or it may simply miss possible local minima by swimming through them without stopping. On the other hand, if the C(i) values are too small, convergence can be slow, but if the search finds a local minimum it will typically not deviate too far from it. $c(i)$ can be treated as a type of "step size" for the optimization algorithm.

3.7.3 Chemotactic step 'N_c': If the size of N_c is chosen to be too short, the algorithm will generally rely more on luck and reproduction, and in some cases, it could more easily get trapped in a local minimum (premature convergence). N_s creates a bias in the random walk (which would not occur if $N_s = 0$), with large values tending to bias the walk more in the direction of climbing down the hill.

3.7.4. Reproduction number 'N_{re}': If N_{re} is too small, the algorithm may converge prematurely; however, larger values of N_{re} clearly increase computational complexity.

3.7.5 Elimination and dispersal number 'N_{ed}': A low value for N_{ed} dictates that the algorithm will not rely on random elimination-dispersal events to try to find favorable regions. A high value increases computational complexity but allows the bacteria to look in more regions to find good nutrient concentrations. Clearly, if p_{ed} is large, the algorithm can degrade to random exhaustive search. If, however, it is chosen appropriately, it can help the algorithm jump out of local optima and into a global optimum.

3.7.6 The parameters that define the cell-to-cell attractant functions 'J_{cc}^{i}':

If the attractant width is high and very deep, the cells will have a strong tendency to swarm (they may even avoid going after nutrients and favor swarming). On the other hand, if the attractant width is small and the depth shallow, there will be little tendency to swarm and each cell will search on its own. Social versus independent foraging is then dictated by the balance between the strengths of the cell-to-cell attractant signals and nutrient concentrations.

3.8 Conclusion

This chapter introduces the concept of the 'Evolutionary Algorithms' with more emphasis on 'Bacterial Foraging'. The basic 'Bacterial Foraging Algorithm' is explained. Chapter-4 describes the implementation of the RBF channel equalizer with 'Bacterial Foraging Algorithm'.

Chapter-4
Channel equalization based on 'Bacterial Foraging Algorithm'

4.1 Introduction

For effective high-speed digital data transmission over a communication channel, the adverse effects of the dispersive channel causing intersymbol interference (ISI), the nonlinearities introduced by the modulation/demodulation process, transmitter and receiver amplifiers and the noise generated in the system are to be suitably compensated. The performance of the linear channel equalizers employing a linear filter with FIR using a least mean square (LMS) or recursive least-squares (RLS) algorithm is limited especially when the nonlinear distortion is severe. In such cases, nonlinear equalizer structures may be conveniently employed with added advantage in terms of lower bit error rate (BER), and higher convergence rate than those of a linear equalizer.

Nonlinear equalizers based on RBF network with bacterial foraging training can perform mapping between its input and output space and are capable of forming decision regions with nonlinear decision boundaries. Because of nonlinear characteristics of the above equalizers and channel equalization being a nonlinear classification problem; these equalizers are best suited for channel equalization problem.

This chapter analyzes the performance of the nonlinear equalizers based on RBF network, with bacterial foraging training. Following the introduction this chapter is organized as follows. Section 4.2 describes calculation of channel states section 4.3 describes the digital transmission system affected by nonlinearities and the channel states of time invariant and time-varying channel. Section 4.4 provides some extensive simulation results.

4.2 Calculation of channel states

To implement the Bayesian equalizer, the concept of channel states is introduced first. The equalizer input vector is defined as,

$r(k) = [r(k), r(k-1),..., r(k-m+1)]^T \in \Re^m$ and $r(k) = \hat{r}(k) + \eta(k)$. The vector $\hat{r}(k)$ is the noise free received signal vector and $\hat{r}(k) = [\hat{r}(k), \hat{r}(k-1),..., \hat{r}(k-m+1) \in \Re^m$.

Each of this possible noise free received signal vectors constitutes a channel state. The channel states are determined by the transmitted symbol vector $s(k) = [s(k), s(k-1),..., s(k-m-n_c+2)]^T$. Here $\hat{r}(k)$ can be represented as $\hat{r}(k) = H[s(k)]$, where matrix $H \in \Re^{m \times (m+n_c-1)}$ is the channel matrix.

$$\mathbf{H} = \begin{bmatrix} a_0 & a_1 & \cdots & a_{n_c-1} & 0 & \cdots & 0 & \cdots & 0 \\ 0 & a_0 & \cdots & a_{n_c-2} & a_{n_c-1} & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \cdots & \cdots & \cdots & a_0 & \cdots & a_{n_c-1} \end{bmatrix} \tag{4.1}$$

Since $s(k)$ has $N_s = 2^{m+n_c-1}$ combinations, $\hat{r}(k)$ has N_s states. These channel states are constructed with N_s sequences of $s(k)$, which can be denoted as,

$$s_j(k) = [s_j(k), s_j(k-1),..., s_j(k-m-n_c+2)]^T \text{ , } 1 \le j \le N_s \tag{4.2}$$

The corresponding channel states are denoted as c_j and are given by

$$c_j = \hat{r}(k) = H[s_j(k)], \ 1 \le j \le N_s \tag{4.3}$$

The channels state matrix, $c_d = \{c_j\}$, $1 \le j \le N_s$ can be partitioned into two subsets depending on the transmitted symbol $s(k-d)$ i.e.,

$$c_d = c_d^{+} U c_d^{-} \tag{4.4}$$

Where,

$$c_d^+ = \left\{ \hat{r}(k) \middle| s(k-d) = +1 \right\}$$

$$c_d^- = \left\{ \hat{r}(k) \middle| s(k-d) = -1 \right\} \quad (4.5)$$

Table.4.1. Channel state calculation for the channel $H(z) = 1 + .5z^{-1}$ with m =2, d =0 and N_s =8.

No:	c_j	S(k)	S(k-1)	S(k-2)	$\hat{r}(k)$	$\hat{r}(k-1)$
1	C_1	-1	-1	-1	-1.5	-1.5
2	C_2	-1	-1	1	-1.5	-0.5
3	C_3	-1	1	-1	-0.5	0.5
4	C_4	-1	1	1	-0.5	1.5
5	C_5	1	-1	-1	0.5	-1.5
6	C_6	1	-1	1	0.5	-0.5
7	C_7	1	1	-1	1.5	0.5
8	C_8	1	1	1	1.5	1.5

4.3 Digital transmission system with nonlinearity in channel

The block diagram of the digital transmission system with equalizer where the Channel is affected by some nonlinearity is presented in Fig.4.1.

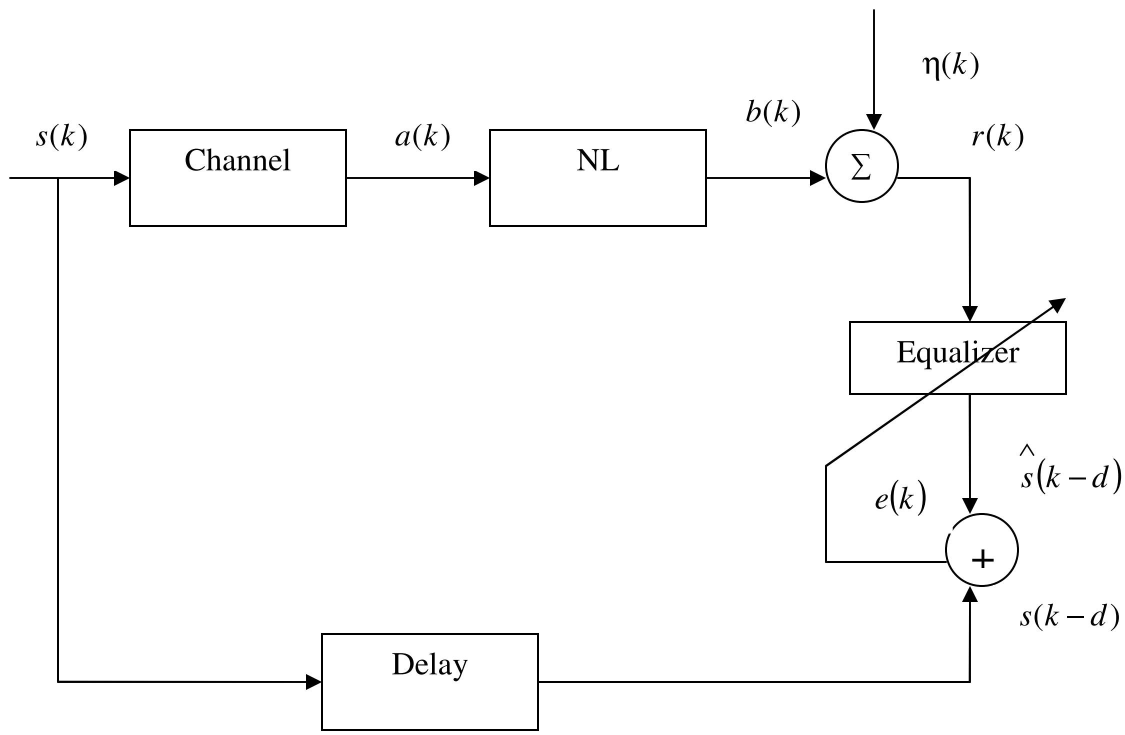


Fig.4.1 Digital transmission system with non linearity and equalizer.

In this baseband structure of the digital communication system with the nonlinearity introduced at the output of the channel, the combined effect of the transmitter filter, the transmission medium, and other components are included in the "channel". A widely used model for a linear dispersive channel is a FIR model whose output at time instant k may be written as

$$a(k) = \sum_{n=0}^{N_c-1} h(n)t(k-n) \tag{4.6}$$

Where $h(n)$ the channel is taps of the channel taps and N_c is the length of the FIR channel model, $a(k)$ is the noise free channel output.

If the nonlinear distortion caused by the channel is to be considered, the channel model is treated as nonlinear and its output may be expressed as

$$b(k) = \varphi(a(k)) \tag{4.7}$$

or

$$a'(k) = \varphi(t(k), t(k-1), \ldots, t(k-N_c)+1); h(0, h(1)), \ldots, h(N_c)\ ) \tag{4.8}$$

Where $\varphi(.)$ is some nonlinear function represented by "**NL**" block shown in Fig 4.1.The channel output is corrupted with AWGN $\eta(k)$ of variance σ^2 to produce $r(k)$, the signal received at the receiver. The purpose of the equalizer is to recover the transmitted symbol $s(k)$ from the knowledge of the received signal symbols without any error. As the channel is affected by some nonlinearity due to either modulation/demodulation or amplifiers used in transmitter and receives, it is difficult to estimate the channel during training period. The scalar channel states can be directly estimated by k-mean clustering method during the transmission of the known data and these scalar states formulate the channel states which forms the centers of the basis function of the equalizers.

In the Fig. 4.1, we have

$$r(k) = \sum_{i=0}^{N} c_i [a(k)]^i + \eta(k) \tag{4.9}$$

Where N denotes the number of nonlinear terms. Considering $a(k) = 1 + .5z^{-1}$ and $c_1 = 1$, $c_2 = 0$, $c_3 = -0.9$, the channel states of this channel $\hat{r}(k)$ can be calculated as shown in Table.4.2.

Table 4.2:Calculation of nonlinearities with channel states.

S.No:	S(k)	S(k-1)	S(k-2)	a(k)	a(k-1)	$\hat{r}(k)$	$\hat{r}(k-1)$
1	-1	-1	-1	-1.5	-1.5	1.5375	1.5375
2	-1	- 1	1	-1.5	-0.5	1.5375	-.3875
3	-1	1	-1	-0.5	0.5	-.3875	.3875
4	-1	1	1	-0.5	1.5	-.3875	-1.5375
5	1	-1	-1	0.5	-1.5	.3875	1.5375
6	1	-1	1	0.5	-0.5	.3875	-.3875
7	1	1	-1	1.5	-0.5	-1.5375	.3875
8	1	1	1	1.5	1.5	-1.5375	-1.5375

4.4 simulation results

4.4.1 Results of a linear channel

Calculation of channel states for the channel $1+.5z^{-1}$ are shown in Table.4.1. These channel states can be represented in a two dimensional space is as shown in Fig.4.2. These channel states are drawn assuming delay as 0.

The signal s is passed through the channel and a noise at an SNR 20db is added to the output of the channel. The received signal vector after the addition of noise is shown in Fig.4.3. The centers of the clusters formed due to the additive noise as shown in Fig.4.3 is calculated using 'Bacterial foraging algorithm', with bacteria population =16 and no: reproduction steps = 8. Positions of the centers after each reproduction step are shown in Fig. 4.4.

An RBF network is designed, with centers as the centers obtained above. The performance of the equalizer can be find out by plotting the BER plot for various SNRs. BER vs SNR plot is shown in Fig 4.5.

Simulated results of the channel $1+.5z^{-1}$, $.5+z^{-1}$ with delays 0, 1 and 2 shown in Chapter-5 and are compared with the results obtained employing LMS algorithm for the same job. Simulations are done for the channels and delays as shown in Table.4.3.

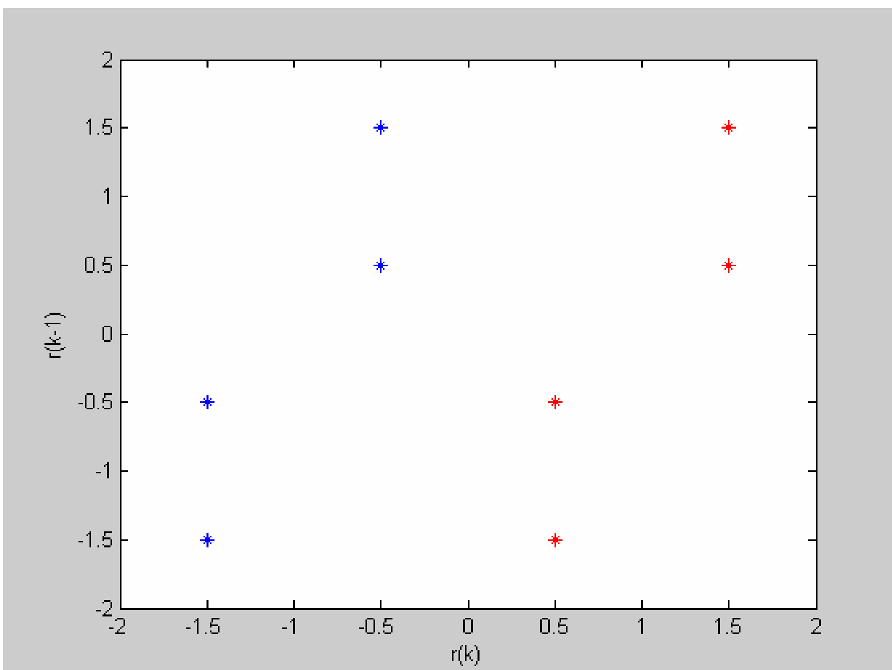


Fig.4.2. channel states for the channel $1+0..5z^{-1}$ with zero delay.

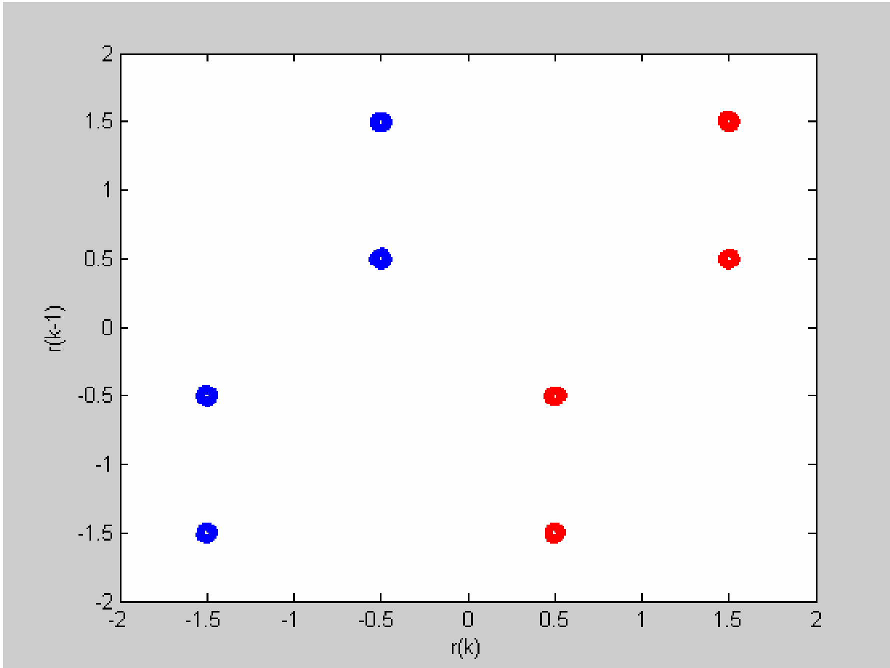


Fig.4.3. Received signal after noise at 20db SNR added

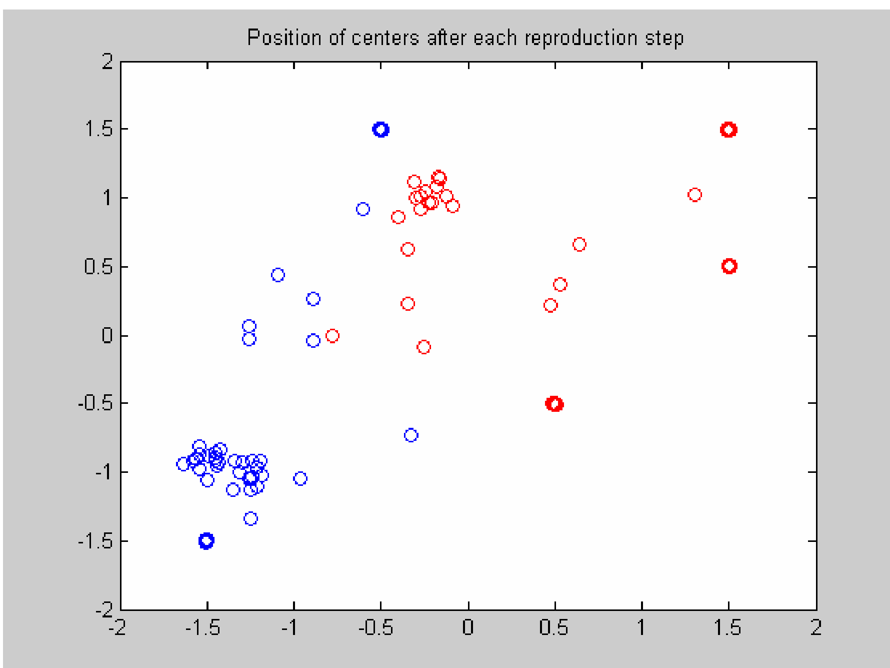


Fig.4.4 position of centers in each generation

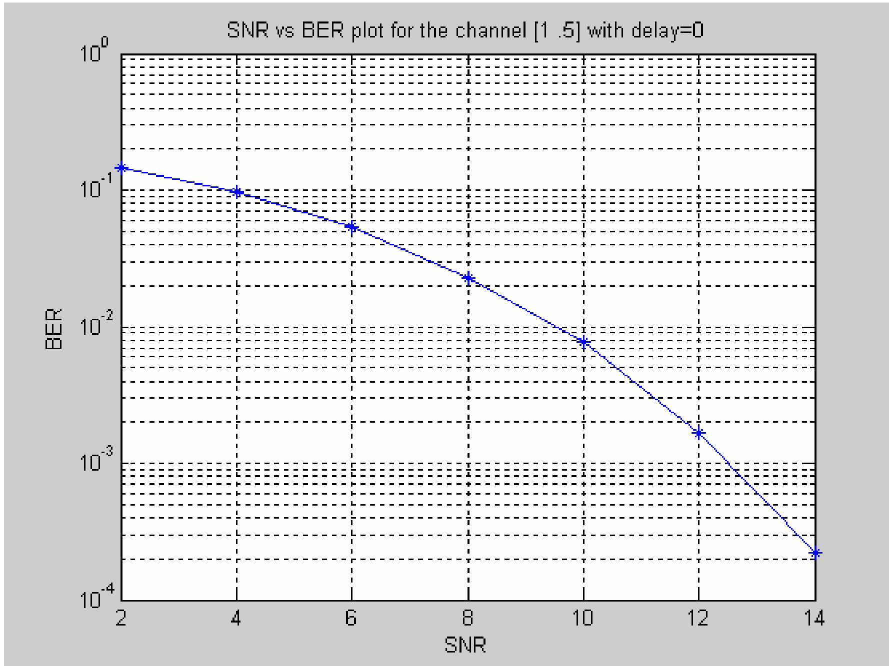


Fig.4.5. SNR vs BER plot for the channel $1+0.5z^{-1}$ *with delay = 0.*

Table.4.3.Linear channels simulated

S.No.	Channel	Simulated for Delays
1	$1+0.5z^{-1}$	0,1,2
2	$0.5+z^{-1}$	0,1,2
3	$0.2682+.09296z^{-1}+0.2682z^{-2}$	0

4.4.2 Results of a non-linear channel

Calculation of channel states for the non-linear channel are shown in Table.4.2. The channel states can be represented in a two dimensional space as shown in Fig.4.6. The delay considered here is 0.

The signal s is passed through the channel $1+.5z^{-1}$ and a non-linearity $a(k)-0.9a(k)^3$. Noise at an SNR 20db is added to the output of the channel. The received signal samples $r(k)$ are shown in Fig.4.7.

The centers of the clusters formed due to the additive noise as shown in Fig.4.7 is calculated using 'Bacterial foraging algorithm', with bacteria population =16 and no: reproduction steps = 16. Positions of the centers after each reproduction step are shown in Fig. 4.8.

An RBF network is designed, with centers as the centers obtained above. The performance of the equalizer can be finding out by plotting the BER plot for various SNRs. BER vs SNR plot is shown in Fig 4.9

Simulated results of other non-linear channels are shown in Chater-5. List of non-linear channels simulated are shown in Table. 4.3.

Table.4.4 Non-linear channels simulated

S.No.	Channel	Non-linearity
1	$1+.5z^{-1}$	$a-.9a^3$
2	$1+.5z^{-1}$	$a+.2a^2-.1a^3$
3	$1+.5z^{-1}$	$\tanh(a)$

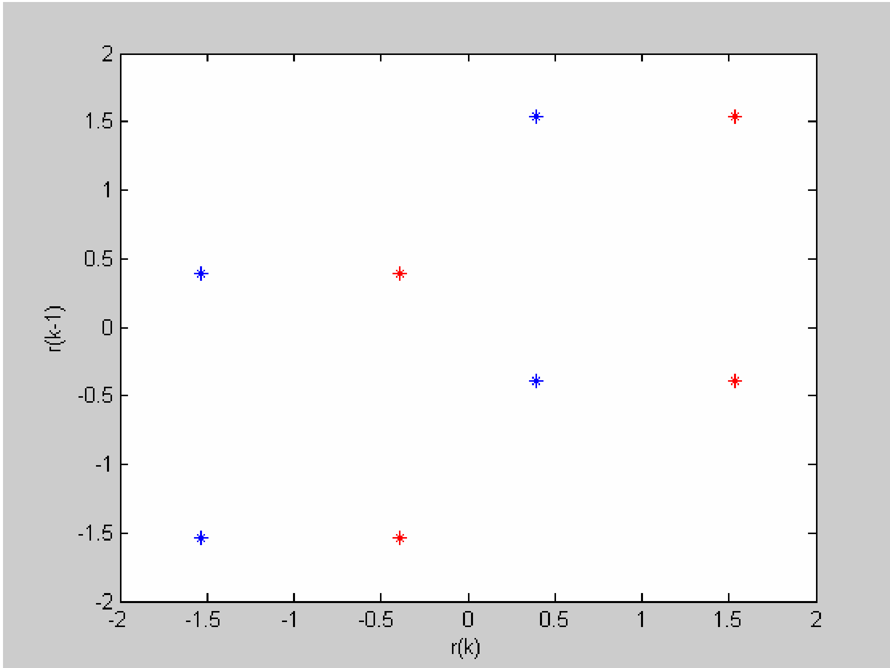


Fig 4.6 channel states for the non-linear channel, with non linearity $a(k)-0.9a(k)^3$

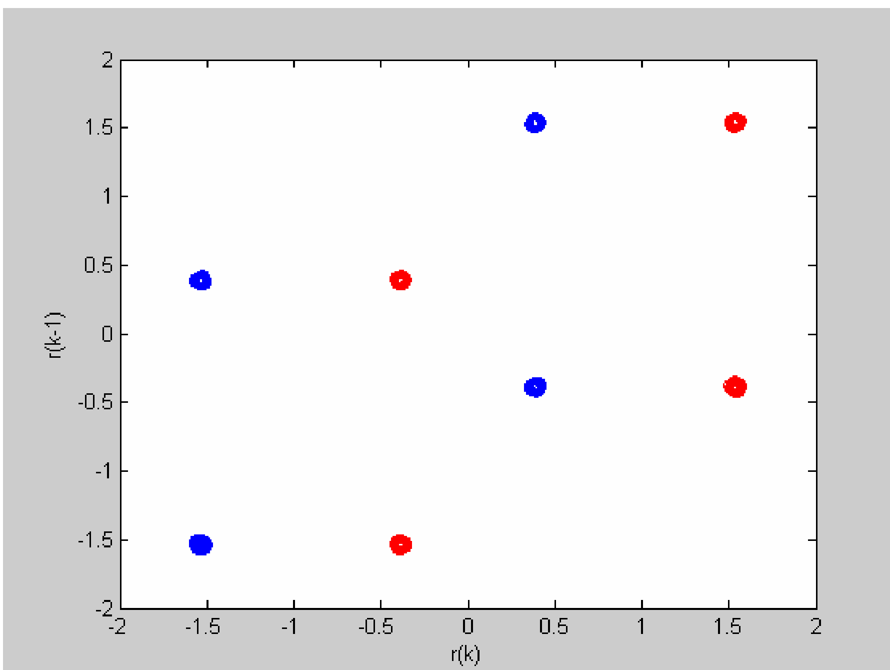


Fig 4.7 channel states for the non-linear channel, with noise at 20db added

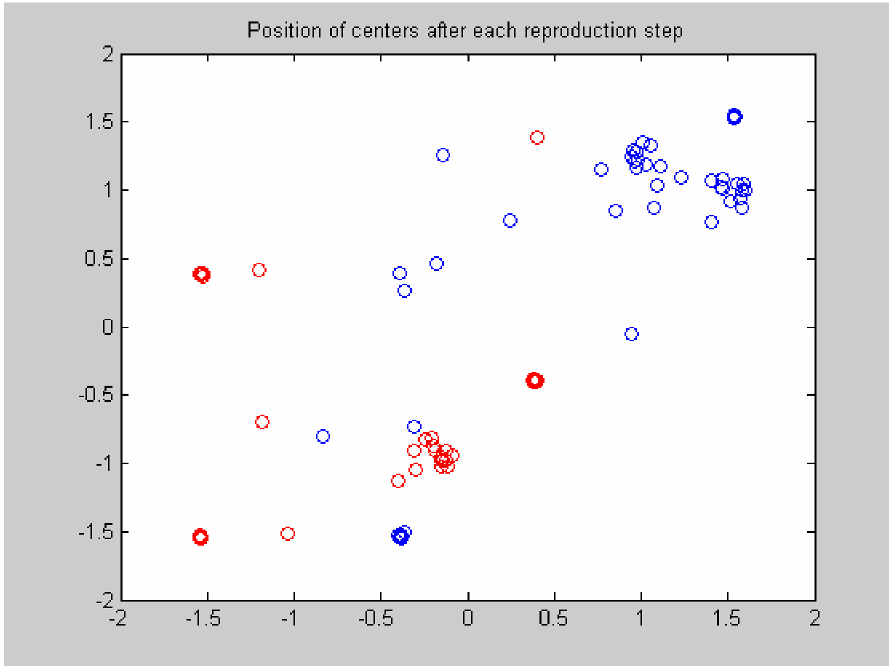


Fig.4.8. position of centers in each generation

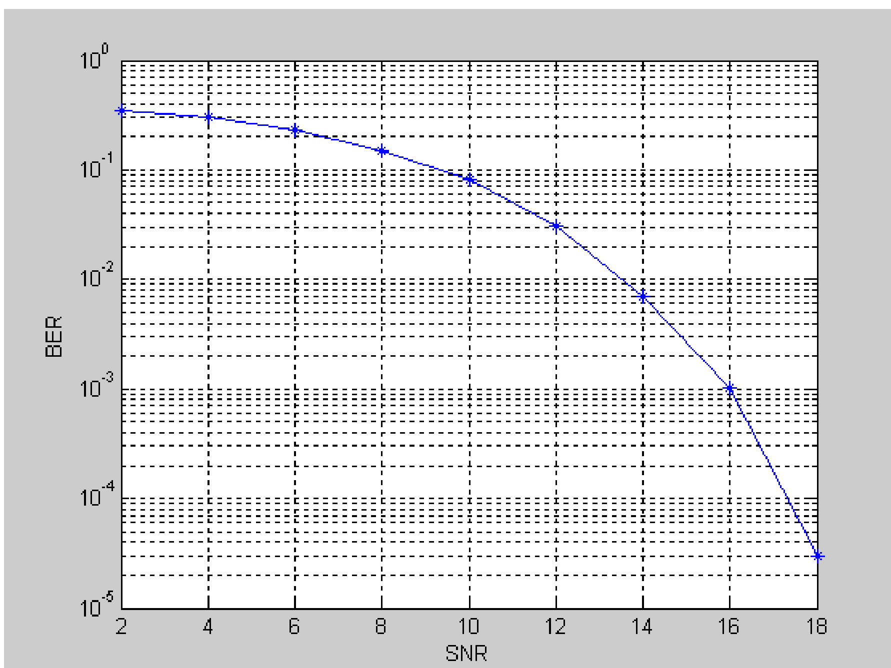


Fig.4.9 performance of the channel $1+.5z^{-1}$ *with non-linearity* $a(k)-0.9a(k)^3$

Chapter-5
Results & Discussions

5.1 Introduction

To validate the performance of the proposed equalizer, the algorithm was tested for channel equalization for a variety of conditions. The simulations were carried out for different types of channels and results obtained with bacterial foraging algorithm was compared with the result obtained with RBF equalizer trained with K-means clustering and linear equalization using LMS algorithm. The simulations were carried out using MATLAB 6.5 and personal computer. The training signal considered is a BPSK signal with symbols $\{\pm 1\}$. These training samples were passed through the channel and AWGN is added to the output of the channel. The resulted signal was given to the equalizer and the equalizer is trained using an RBF network with bacterial foraging algorithm.

This chapter is organized as follows: following this introduction section 5.2 presents the simulation results for minimum phase channel. Section 5.3 presents the simulation results for the non minimum phase channel. Sections 5.4 and 5.5 presents the results for mixed phase and non-linear channels respectively. Finally section 5.6 discusses the performance of the proposed equalizer.

5.2 Simulation results for minimum phase channel

First simulation was conducted for a minimum phase channel whose transfer function is $1+0.5z^{-1}$. This channel has only one zero and it is inside the unit circle in the 2-dimensional plane.

The equalizer order used was 2. The equalizer was trained with 1000 samples using bacteria population 16 over 8 reproduction steps. Number of chemotactic steps was set to the number of members in the particular cluster. The channel states estimated using bacterial foraging algorithm were compared with the actual centers as shown in Table 5.1 The equalizer performance in terms of BER is compared with RBF equalizer trained with K-means clustering and LMS equalizer. The BER for varying SNR is plotted in Fig 5.1a through 5.1c for decision delay of 0,1 and 2 respectively.

Table 5.1 Centers obtained for the channel $1+0.5z^{-1}$

	Centers obtained using bacterial foraging		Actual centers	
	r(k)	r(k-1)	r(k)	r(k-1)
c_1	-1.5006	-1.4971	-1.5	-1.5
c_2	-1.5023	-0.5014	-1.5	-0.5
c_3	-0.5001	0.4965	-0.5	0.5
c_4	-0.5032	1.5013	-0.5	1.5
c_5	0.4986	-1.5025	0.5	-1.5
c_6	0.5012	-0.5025	0.5	-0.5
c_7	1.4992	0.4968	1.5	0.5
c_8	1.5037	1.4998	1.5	1.5

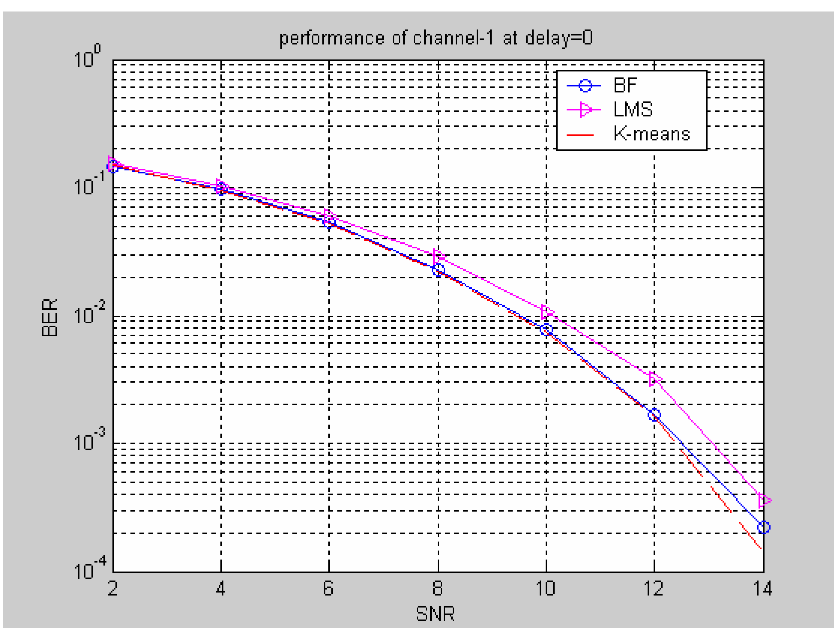


Fig.5.1a. performance of the equalizer at delay=0.

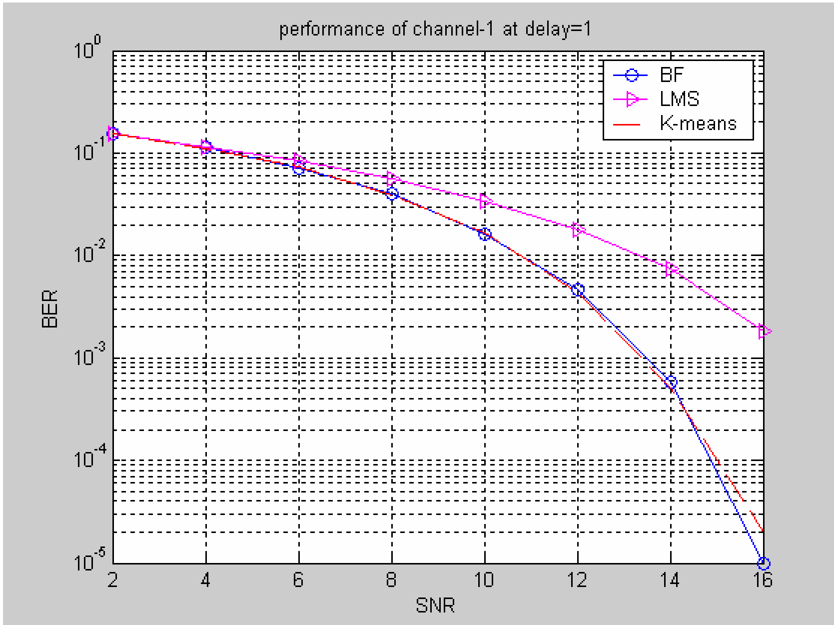
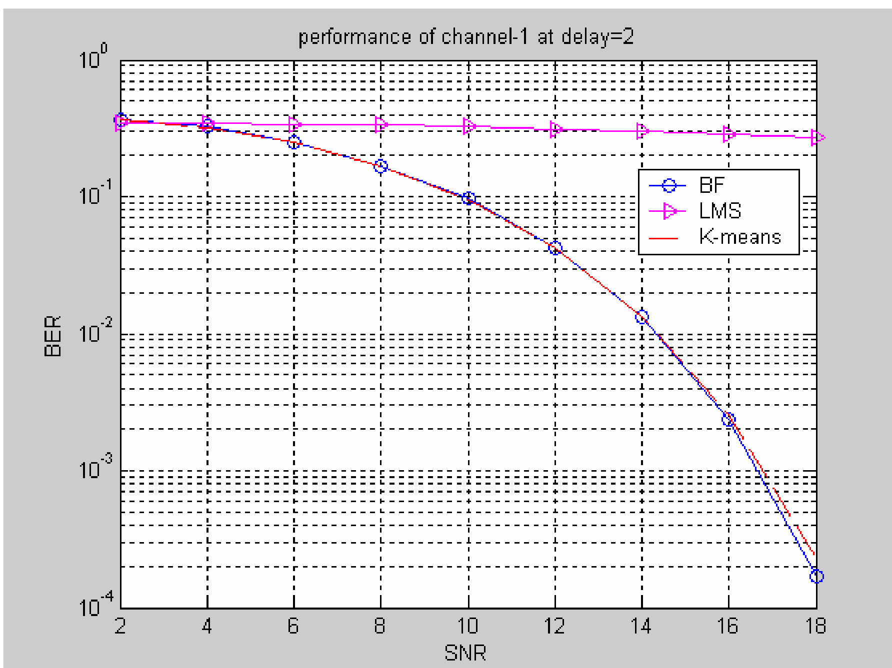


Fig.5.1b. performance of the equalizer at delay=1.



5.1c. performance of the equalizer at delay=2.

From the plots it is seen that at delay = 0 and 1 the proposed equalizer performs better than LMS equalizer even though the problem is linearly sepearble. But at delay=2 the equalization problem becomes linearly inseparable. In this case LMS algorithm completely fails to classify the patterns. The proposed equalizer gains almost 4db SNR at a BER of10^{-3}. In all the three cases the proposed equalizer performs close to the RBF equalizer trained with K-means clustering.

5.3 Simulation results for non-minimum phase channel

Second simulation was conducted for a non-minimum phase channel whose transfer function is $0.5 + z^{-1}$. The only zero of this channel is located outside the unit circle.
The parameters taken for the simulation are same as those taken for a minimum phase channel.The channel states estimated using bacterial foraging algorithm were compared with the actual centers as shown in Table 5.2. The equalizer performance in terms of BER is compared with RBF equalizer trained with K-means clustering and LMS equalizer in Fig 5.2a through 5.2c for varying SNR for decision delays 0,1 and 2 respectively.

Table 5.2 Centers obtained for the channel $0.5 + z^{-1}$

	Centers obtained using bacterial foraging		Actual centers	
	r(k)	r(k-1)	r(k)	r(k-1)
c_1	-1.5006	-1.4971	-1.5	-1.5
c_2	-1.5023	0.5007	-1.5	0.5
c_3	0.5026	-0.5004	0.5	-0.5
c_4	0.4959	1.5003	0.5	1.5
c_5	-0.5035	-1.5006	-0.5	-1.5
c_6	-0.5039	0.4982	-0.5	0.5
c_7	1.4984	-0.5017	1.5	-0.5
c_8	1.5037	1.4998	1.5	1.5

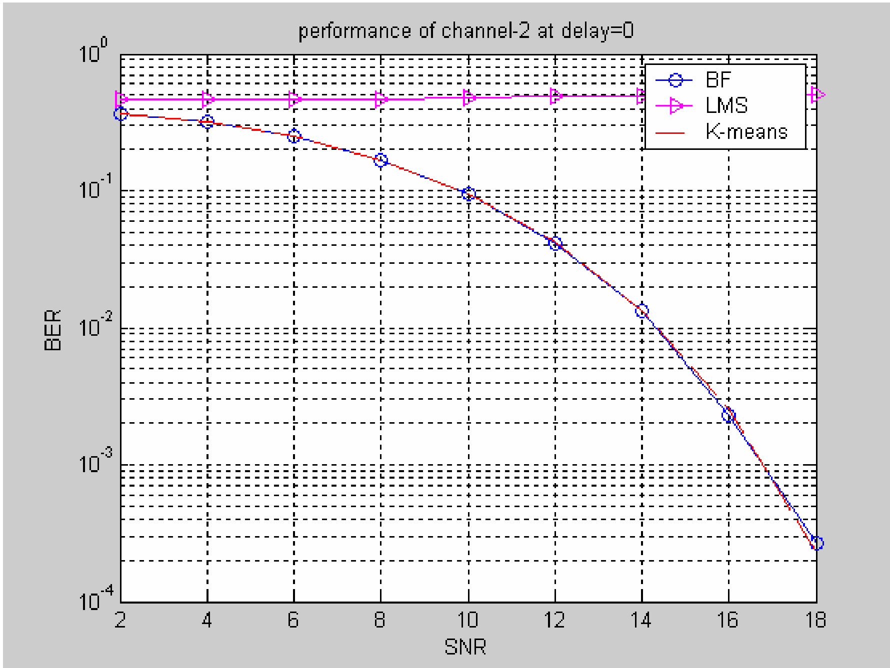


Fig.5.2a. performance of the equalizer with delay=0

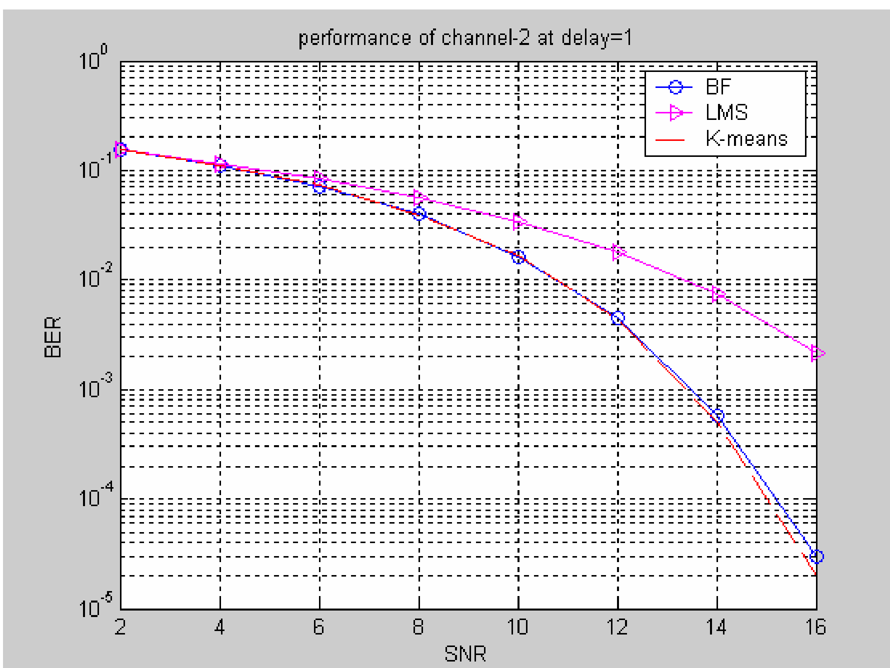


Fig.5.2b. performance of the equalizer with delay=1

.

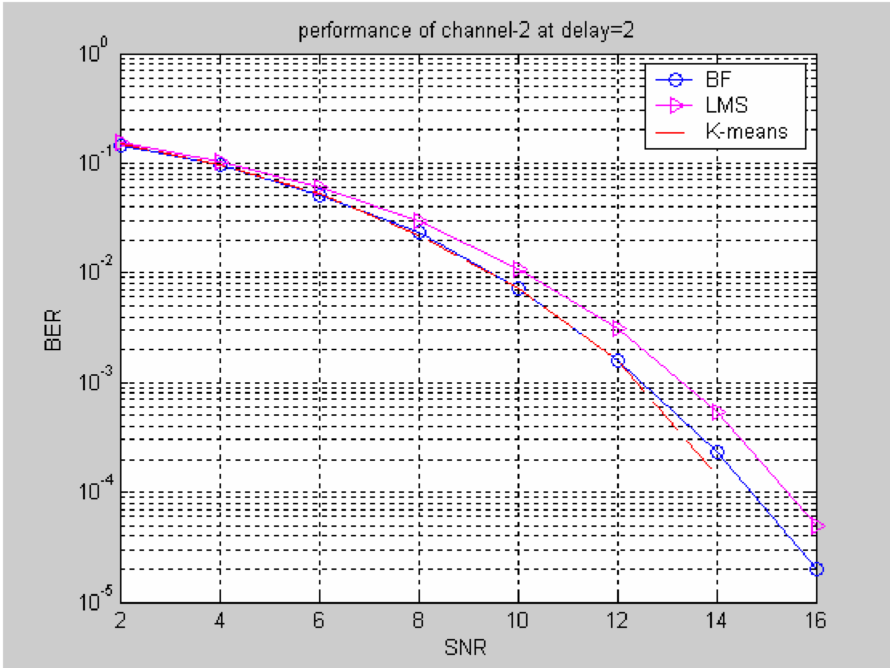


Fig.5.2c. performance of the equalizer with delay=2

From the plots it is seen that at delay=0 the proposed equalizer gains SNR of approximately 4db (Fig 5.2a), where the equalization problem becomes linearly inseparable. In Fig 5.2b and 5.2c the proposed equalizer performs better than the LMS equalizer even though the problem is linearly seperable.In all the three cases the 'Bacterial Foraging Algorithm' performs almost similar to K-means clustering.

5.4 simulation results for a mixed phase channel

Second simulation was conducted for a non-minimum phase channel whose transfer function is $0.2682+0.9296z^{-1}+0.2682z^{-2}$. Of the two zeros one zero lies inside the unit circle and the other one lies out side the unit circle.

The parameters taken for the simulation are same as those taken in sections 5.2 and 5.3. The channel states estimated using bacterial foraging algorithm were compared with the actual centers as shown in Table 5.3. The equalizer performance in terms of BER is compared with LMS equalizer in Fig 5.3 for varying SNR for decision delay 0.

Table 5.3 Centers obtained for the channel $0.2682 + 0.9296z^{-1} + 0.2682z^{-2}$

	Centers obtained using 'Bacterial Foraging Algorithm'		Actual Centers	
	r(k)	r(k-1)	r(k)	r(k-1)
c_1	-1.466	-1.466	-1.4680	-1.4660
c_2	-1.466	-0.9296	-1.4679	-0.9301
c_3	-0.9296	0.3932	-0.9334	0.3945
c_4	-0.9296	0.9296	-0.9283	0.9292
c_5	0.3932	-0.9296	0.3911	-0.9292
c_6	0.3932	-0.3932	0.3907	-0.3920
c_7	0.9296	0.9296	0.9292	0.9292
c_8	0.9296	1.466	0.9298	1.4641
c_9	-0.9296	-1.466	-0.9289	-1.4677
c_{10}	-0.9296	-0.9296	-0.9310	-0.9298
c_{11}	-0.3932	0.3932	-0.3939	-0.3920
c_{12}	-0.3932	0.9296	-0.3998	0.9317
c_{13}	0.9296	-0.9296	0.9303	-0.9300
c_{14}	0.9296	-0.3932	0.9309	-0.3959
c_{15}	1.466	0.9296	1.4637	0.9269
c_{16}	1.466	1.466	1.4668	1.4641

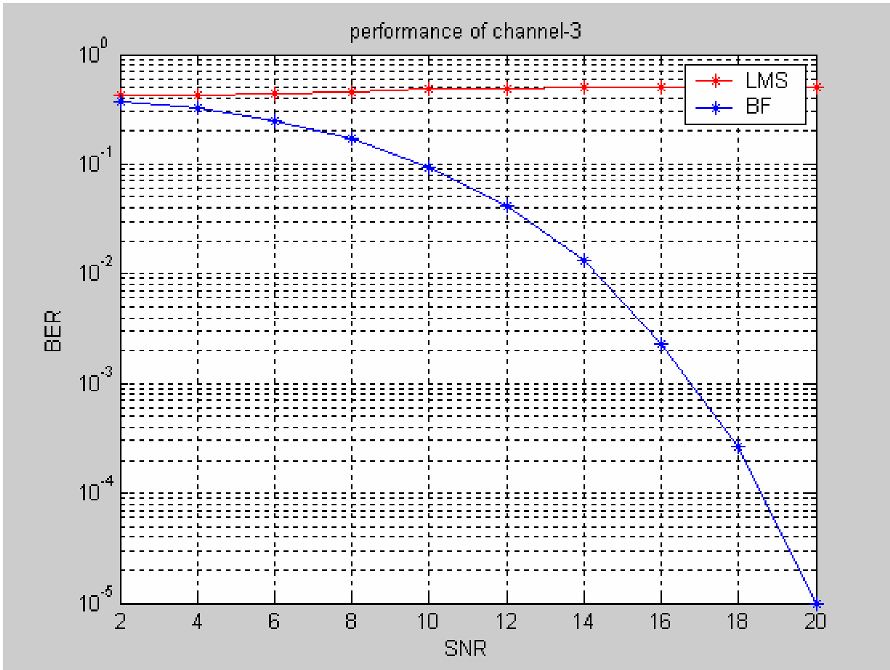


Fig.5.3 performance of the equalizer at delay=0

For the channel $0.2682 + 0.9296z^{-1} + 0.2682z^{-2}$, at delay =0 the equalization problem becomes linearly inseparable. In this case LMS equalizer fails to solve the problem, whereas the proposed equalizer performs well under this non-linear condition.

5.5 Simulation results for non-linear channels

Final simulation was carried on various non-linear channels shown in Table 5.4. training parameters chosen are same as those in the previous sections. The equalizer performance in terms of BER is compared with LMS equalizer in Fig 5.4 for varying SNR

Table.5.4 Various non-linearities considered for the simulation

NL#1	$a(k)\ a(k) - 0.9a(k)^3$
NL#2	$a(k) + 0.2a(k)^2 - 0.1a(k)^3$
NL#3	$\tanh(a(k))$
NL#4	$a(k) - 0.9a(k)^3$

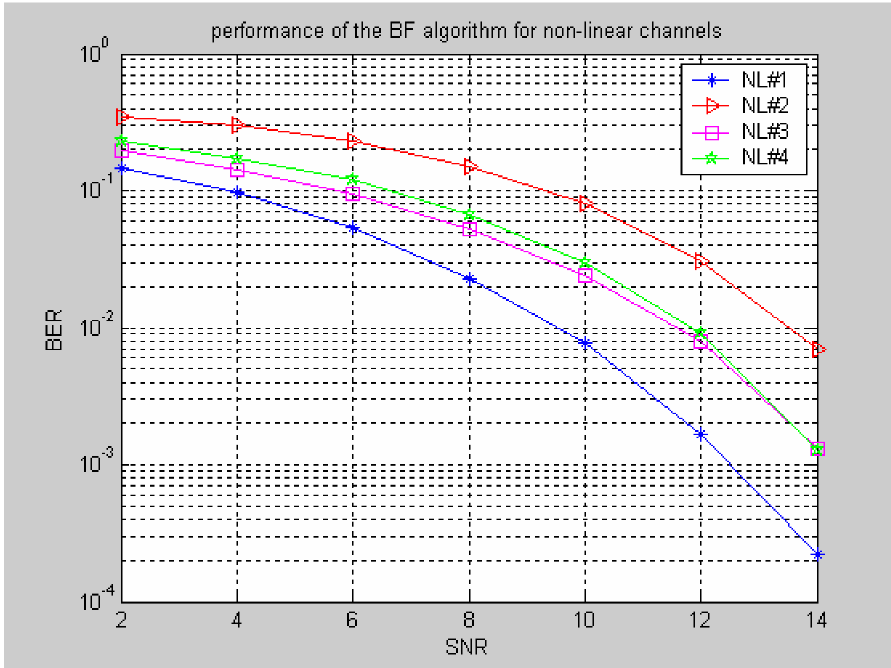


Fig. 5.4 performance of the equalizer for various non-linear channels
(The no-linear channels are shown in Tabl.5.1)

5.6 Discussion:

From the above sections it can be seen that that the proposed equalizer performs similar to an RBF equalizer trained with K-means clustering. It can also be seen that the proposed equalizer gives good results when the problem becomes linearly inseparable, where the LMS equalizer fails to solve the equalization problem. The proposed equalizer is expected to give much better results for higher order of the equalizer [25].

Chapter-6
Conclusion

6.1 Achievement of the thesis work

This thesis work proposes a new technique for channel equalization. Proposed equalizer has been tested for both linear and non-linear equalizers and the performance of the equalizer is compared with the conventional LMS algorithm and RBF based equalizer as well. Proposed equalizer performs closed to optimal when the channel becomes non-linear. The proposed equalizer is expected to give much better results for higher equalizer orders.

6.2 Scope of future work

- As the bacterial foraging algorithm is working well for the channel equalization in digital communication, it can be applied to design efficient mobile communication receiver, which can mitigate the effects of multipath losses.
- The algorithm trains the system like a biological system, it can be applied to solve the blind equalization problem.
- The algorithm may be tested for QPSK and other modulation techniques.
- The algorithm may be applied to mobile LAN where length of training samples is small.
- The equalizer may be applied to other communications like satellite communication.
- Other 'Evolutionary Algorithms' like 'Ant Colony Optimization' may be applied to solve the channel equalization problem.

References

[1] T. S. Rappaport, "Wireless Communications: Principles and Practice", ***Pearson Education, 2002.***

[2] R. W. Lucky, "Automatic Equalization of Digital Communication", ***Bell system Tech. J, vol. 44, pp. 547-588, April 1965.***

[3] G. D. Forney, "Maximum-Likelihood Sequence estimation of Digital Sequence in the presence of Intersymbol Interference", ***IEEE Transaction on Information Theory, vol. IT-18, pp. 363-378, May 1972.***

[4] G. D. Forney, "The Viterbi Algorithm", ***Proceedings of the IEEE, vol. 61, pp. 268-278, March 1973.***

[5] F. R. Magee Jr and J. G. Proakis, "Adaptive Maximum-Likelihood Sequence Estimation for Digital Signaling in the presence of Intersymbol Interference", ***IEEE Transaction on Information Theory, vol. IT-19, pp. 120-124, January 1973.***

[6] D. S. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks", ***Complex Systems, vol. 2, pp. 321-355,1998.***

[7] D. R. Hush and B. G. Horne, "Progress in Supervised Neural Networks: Whats New Since Lippmann?", ***IEEE Signal Processing Magazine, vol. 10, January 1993.***

[8] M. J. D. Powell "Radial basis function for multivariable interpolation: A review", in ***Algorithms for Approximation of Functions and Data (J. C. Mason and M. G. Cox, eds.), pp. 143-167, Oxford University press, 1987.***

[9] S. Haykin, *Neural Networks – A Comprehensive Foundation*. New York: ***Macmillan, 1994.***

[10] D. Godard, "Channel Equalization Using Kalman Filter for Fast Data Transmission", ***IBM Journal Res. Development, vol. 18, pp. 267-273, May 1974.***

[11] L. X. Wang and J. M. Mendel, "Fuzzy Adaptive Filters, with Application to Non-Linear Channel Equalization", ***IEEE Transactions of Fuzzy Systems, vol. 1, pp. 161-170, August 1993.***

[12] ***http://www.cbhd.org/resources/biotech/tongen_2003-11-07_print.htm***

[13] Mubarak M. S. Al-Sawafi, Joseph A. Jervase, "A Micro-Genetic Algorithm-based CDMA Multi-user Detector",***Proceedings of the Second Annual Conference on Communication Networks and Services Research.***

[14] S. Chen, Y. Wu, and S. McLaughlin, "Genetic Algorithm Optimization for Blind ChannelIdentification with Higher Order Cumulant Fitting", **IEEE Transactions On Evolutionary Computation, Vol. 1, No. 4, November 1997, pp.259-265.**

[15] Dean Joseph Krusienski, "Enhanced Structured Stocastic Global Optimization Algorithms For Iir And Nonlinear Adaptive Filtering", ***PhD thesis submitted for the partial fulfillment of the award of doctor of philosophy., Aug 2004, The Pennsylvania State University.***

[16] S. Mishra "A Hybrid Least Square-Fuzzy Bacterial Foraging Strategy For Harmonic Estimation", ***IEEE Transactions On Evolutionary Computation, Vol. 9, No. 1, February 2005, pp 61-73.***

[17] B. Widrow and M. E. Hoff(jr), "Adaptive switching circuits", ***IRE WESCON Conv., vol. 4, pp. 94-104, August 1960.***

[18] D. A. George, R. R. Bowen, and J. R. Storey" An Adaptive Decision Feedback Equalizer", ***IEEE Transactions on Communication Technology, vol. COM-19, pp. 281-293, June 1971.***

[19] S. U. H. Qureshi, "Adaptive Equalization" ***Proceeding of the IEEE, vol.73, pp. 1349-1387, September 1985.***

[20] G. J. Gibson, S. Siu, and C. F. N. Cowan, "The Application of Nonlinear Structures to the Reconstructed of Binary Signals", ***IEEE Transactions on Signal Processing, vol. 39, pp. 1877-1884, August 1991.***

[21] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal Least Squares Learning Algorithms for Radial Basis Function Networks", ***IEEE Transactions on Neural Networks, vol. 2, pp. 302-309, March 1991.***

[22] S. Chen, G. J. Gibson, C. F. N. Cowan, and P. M. Grant, "Reconstructed of Binary Signals using an Adaptive Radial Basis Function Equalizer", ***Signal Processing (Eurasip), vol. 22, pp. 77-93, January 1991.***

[23] B. Mulgrew, ”Nonlinear Signal Processing for Adaptive Equalization and Multiuser Detection”, in ***Proceeding of the European Signal Processing Conference, EUSIPCO, (Island of Rhodes, GREECE), pp. 537-544, 8-11 September 1998.***

[24] J. Cid-Sueiro and A. R. Figueiras-Vidal, ”Channel Equalization with Neural Networks”, in ***Digital Signal Processing in Telecommunications – European Project COST#229 Technical Contributions (A. R. Figueiras – Vidal, ed.), pp. 257-312, London, U.K.: Springer-Verlag, 1996.***

[25] B. Mulgrew, ”Applying Radial Basis Functions”, ***IEEE Signal Processing Magazine, vol. 13, pp. 50-65, March 1996.***

[26] [31] J. G. Proakis, *Digital communication*, ***McGraw-Hill, 3rd edition, New York, 1995.***

[27] B. R. Peterson and D. D. Falconer, “Minimum Mean Square Equalization in Cyclostationary and Stationary Interference-Analysis and Subscriber Line Calculations”, ***IEEE Journal on Selected Areas in Communication, vol. 9, pp. 931–940, August 1991.***

[28] B. R. Peterson and D. D. Falconer, “Minimum Mean Square Equalization in Cyclostationary and Stationary Interference-Analysis and Subscriber Line Calculations”, ***IEEE Journal on Selected Areas in Communication, vol. 9, pp. 931–940, August 1991.***

[29] J. H. Winter, “Optimum Combining in Digital Mobile Radio with Cochannel Interference”, ***IEEE Journal on Selected Areas in Communication, vol. SAC-2, pp. 528–539, July 1984.***

[30] J. G. Proakis, “Adaptive Equalization for TDMA Digital Mobile Radio”, ***IEEE Transactions on Vehicular Technology, vol. 40, pp. 333–341, May 1991.***

[31]K. Feher, “MODEMS for Emerging Digital Cellular-Mobile Radio system”, ***IEEE Transactions on Vehicular Technology, vol. 40, pp. 355–365, May 1991.***

[32] B. R. Peterson and D. D. Falconer, “Supression of Adjacent Channel Interference in Digital Radio by Equalization”, in ***Proceeding of IEEE International Conference on Communications, (Chicago, IL, USA), pp. 0657–0661, IEEE, 14-18 June 1992.***

[33] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. ***John Wiley and Sons, 1973.***

[34] S. Chen, B. Mulgrew, and S. McLaughlin, "Adaptive Bayesian Equalizer with Decision Feedback", ***IEEE Transactions on Signal Processing, vol. 41, pp. 2918–2927, September 1993.***

[35] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control", ***IEEE. Control Syst. Mag., pp. 52–67, Jun. 2002***.

[36] S.K Patra ,**'**Development of Fuzzy System Based Channel Equalizers' ***PhD thesis submitted for the partial fulfillment of the award of doctor of philosophy.,*** **The University of Edinburgh**. **August 1998**